

# Sun™ MediaCenter™ Server Administrator's Guide

---

*Release 2.1*



THE NETWORK IS THE COMPUTER™

A Sun Microsystems, Inc. Business  
901 San Antonio Road  
Palo Alto, CA 94303 USA  
415 960-1300 fax 415 969-9131

Part No.: 805-3194-10  
Revision A, December 1997

© 1997 Sun Microsystems, Inc. All rights reserved.  
901 San Antonio Road, Palo Alto, California 94303 U.S.A.

This product and related documentation are protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or related documentation may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Portions of this product may be derived from the UNIX<sup>®</sup> and Berkeley 4.3 BSD systems, licensed from UNIX System Laboratories, Inc., a wholly owned subsidiary of Novell, Inc., and the University of California, respectively. Third-party font software in this product is protected by copyright and licensed from Sun's font suppliers.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the United States Government is subject to the restrictions set forth in DFARS 252.227-7013 (c)(1)(ii) and FAR 52.227-19.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

#### TRADEMARKS

Sun, the Sun logo, Sun Microsystems, Solaris, Sun Media Center, SunATM, SunFastEthernet, Java, and HotJava are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and certain other countries. UNIX is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd. OPEN LOOK is a registered trademark of Novell, Inc. PostScript and Display PostScript are trademarks of Adobe Systems, Inc. Mpeg Expert and Applied Vision are trademarks of Applied Vision. All other product names mentioned herein are the trademarks of their respective owners.

All SPARC trademarks, including the SCD Compliant Logo, are trademarks or registered trademarks of SPARC International, Inc. SPARCstation, SPARCserver, SPARCengine, SPARCstorage, SPARCware, SPARCclassic, SPARCcluster, SPARCdesign, SPARC811, SPARCprinter, UltraSPARC, microSPARC, SPARCworks, and SPARCcompiler are licensed exclusively to Sun Microsystems, Inc. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK<sup>®</sup> and Sun<sup>™</sup> Graphical User Interfaces were developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

X Window System is a trademark of the X Consortium.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE PUBLICATION. SUN MICROSYSTEMS, INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME.



# Contents

---

## **1. Introduction 1-1**

- 1.1 Introduction 1-1
- 1.2 Sun MediaCenter Hardware 1-2
- 1.3 Sun MediaCenter Software 1-4
  - 1.3.1 Media File System (MFS) 1-5
  - 1.3.2 Media Stream Manager 1-7
  - 1.3.3 Content Manager 1-8
  - 1.3.4 Web-based Tools 1-8
- 1.4 Video Output Interfaces 1-8
  - 1.4.1 ATM 1-9
  - 1.4.2 LAN 1-9
- 1.5 Content Preparation and Loading 1-10
- 1.6 Server Management 1-11
- 1.7 Push Model 1-11
- 1.8 Design Benefits 1-12

## **2. Preparing and Loading Content 2-1**

- 2.1 General MPEG Encoding Requirements 2-1
- 2.2 MPEG Encoding Requirements for Trick Play 2-2

2.3	Splice Points	2-3
2.3.1	Packet Boundary	2-3
2.3.2	closed_gop	2-4
2.3.3	payload_unit_start_indicator	2-4
2.3.4	adaptation_field_control	2-4
2.3.5	random_access_indicator	2-4
2.3.6	PCR_flag	2-4
2.4	Content Loading	2-5
2.5	References	2-7
<b>3.</b>	<b>Monitoring the Server</b>	<b>3-1</b>
3.1	Overview	3-1
3.2	Using the WebAdmin Tool	3-2
3.2.1	Invoking the Tool	3-2
3.2.2	Common Page Elements	3-5
3.3	Resource Utilization	3-5
3.4	Streaming Videos	3-6
3.5	Copy Sessions	3-9
3.6	Disk Capacity	3-12
3.7	Troubleshooting	3-15
3.7.1	Disk Health	3-17
3.7.2	Console Log	3-18
3.7.3	Scheduler and Bitpump	3-20
<b>4.</b>	<b>Moving Content</b>	<b>4-1</b>
4.1	Overview	4-1
4.2	Using CmTool	4-2
4.2.1	Invoking the Tool	4-2
4.2.2	Common Page Elements	4-5
4.3	Loading Content	4-5

4.4	Table of Contents Page	4-12
4.4.1	View Menu	4-13
4.4.2	Selected Menu	4-15
<b>5.</b>	<b>Sun MediaCenter Server Utilities</b>	<b>5-1</b>
5.1	Setting a Title Access Control List	5-1
5.2	The <code>smc_copy</code> Utility	5-4
5.3	The <code>smc_tar</code> Utility	5-8
5.3.1	Syntax and Options	5-8
5.3.2	Content Loading	5-9
5.3.3	Copying Content	5-10
5.3.4	Remote Copying and Extraction	5-12
5.3.5	Examples	5-12
5.4	The <code>smc_ls</code> Utility	5-17
5.5	The <code>smc_rm</code> Utility	5-21
<b>6.</b>	<b>Sun MediaCenter Server FTP Daemon</b>	<b>6-1</b>
6.1	Keyword for Video-File Functions	6-3
6.2	Video-File Attributes	6-4
6.3	Video-File Commands	6-5
6.3.1	Listing Files	6-7
6.3.2	Moving Files	6-9
6.3.3	Renaming Files	6-13
6.3.4	Deleting Files	6-14
6.4	Loading Content with <code>ftp</code>	6-14

<b>7.</b>	<b>Server Output Data Format</b>	<b>7-1</b>
7.1	Introduction	7-1
7.2	ATM Encapsulation	7-2
7.2.1	MPEG-2 Transport Stream Packet Encapsulation	7-2
7.2.2	MPEG-2 Program Stream Packet Encapsulation	7-3
7.2.3	MPEG-1 System Stream Packet Encapsulation	7-3
7.2.4	Thomson Electronics Packet Encapsulation	7-3
7.2.5	MPEG-2 Encapsulation over Classical IP over AAL5	7-3
7.3	ATM Addressing	7-4
7.3.1	MPEG over AAL5	7-4
7.3.2	MPEG-2 over Classical IP over AAL5	7-4
7.4	Fast Ethernet Encapsulation	7-5
7.4.1	MPEG-2 Transport Stream Packet Encapsulation	7-6
7.4.2	MPEG-2 Program Stream Packet Encapsulation	7-7
7.4.3	MPEG-1 System Stream Packet Encapsulation	7-7
7.4.4	Thomson Electronics Encapsulation	7-7
7.5	Fast Ethernet Addressing	7-7
7.6	Specifying Encapsulation	7-8
7.7	Differences Among Different Types of Links	7-9
7.8	References	7-9
<b>8.</b>	<b>Administering MFS Disks</b>	<b>8-1</b>
8.1	Monitoring MFS Disks	8-1
8.2	MFS Utilities	8-2
8.2.1	<code>mfs df</code>	8-2
8.2.2	<code>mfs diskusg</code>	8-4
8.2.3	<code>mfs repair</code>	8-5
8.2.4	<code>mkmfs</code>	8-8
8.3	Replacing a Failed Disk	8-10

## **9. Access Control Lists 9-1**

- 9.1 ACL Files 9-1
- 9.2 Permissions 9-2
- 9.3 Server ACL 9-2
- 9.4 Player ACL 9-3
- 9.5 Title ACL 9-4
- 9.6 Session ACL 9-4

## **10. Troubleshooting 10-1**

- 10.1 Symptoms and Possible Solutions/Workarounds 10-1
- 10.2 Error Messages 10-4
  - 10.2.1 Server Console Messages 10-4
  - 10.2.2 MFS Error Messages 10-5

### **A. Setting up the SNMP Agent A-1**

- A.1 The Sun MediaCenter Server MIB A-2
- A.2 Enabling Management by Solstice Domain Manager A-6
- A.3 Enabling Trap Handling A-9

### **B. Creating a Content Package B-1**

- B.1 Content Package B-2
- B.2 The TOC File B-3
  - B.2.1 ContentObjects B-5
  - B.2.2 BitStreamObjects B-6
  - B.2.3 DataFileObjects B-7
- B.3 Simplest Case TOC File B-8
- B.4 TOC File for Large Files B-9
- B.5 Index File Requirements B-10
  - B.5.1 Normal Play Time B-11
  - B.5.2 File Offset B-12

B.6 Tape Storage Format B-12

B.6.1 Tar Format B-12

B.6.2 Tapes B-14

B.7 Example B-14

B.8 References B-18

**Glossary Glossary-1**

**Index Index-1**



# Preface

---

This document, the *Sun MediaCenter Server Administrator's Guide*, tells you how to set up, load content onto, and perform other administrative duties relevant to a Sun™ MediaCenter™ server. Specifically, this document describes:

- at a high level, the hardware and software architecture of the Sun MediaCenter server;
- the content format required by the Sun MediaCenter server, plus utilities to load and backup that content;
- the format of the data output by the Sun MediaCenter server;
- how to configure the SNMP agent shipped with the Sun MediaCenter for use with Solstice™ Domain Manager™ or another SNMP-capable network manager.

---

## Who Should Use This Book

This manual is intended for those who administer the Sun MediaCenter server.

We expect the reader to have experience in Solaris® network and system administration and be familiar with maintenance issues related to servers.

---

# The Sun MediaCenter Documentation Suite

This manual is a companion to the *Sun MediaCenter Server Installation and Configuration Manual*, the *Sun MediaCenter Server Programmer's Guide* and the *Sun MediaCenter UltraSPARC Hardware Manual*. Together these manuals are complementary of the hardware manuals shipped with the Sun products that make up a Sun MediaCenter server. These documents include manuals for the server, for the network interface (such as SunFastEthernet™ documentation), and the disk subsystem.

---

## How This Book Is Organized

**Chapter 1 “Introduction,”** introduces the Sun MediaCenter software at a high-level.

**Chapter 2 “Preparing and Loading Content,”** specifies the Sun MediaCenter server's requirements for MPEG content. Trick-play format is defined here.

**Chapter 3 “Monitoring the Server,”** describes the WebAdmin tool, a Web-based interface for monitoring a Sun MediaCenter server.

**Chapter 4 “Moving Content,”** describes the CmTool, a Web-based interface for transferring content to a Sun MediaCenter server.

**Chapter 5 “Sun MediaCenter Server Utilities,”** describes the use of `smc_copy` and `smc_tar`, used for moving content, as well as `smc_ls` and `smc_rm`.

**Chapter 6 “Sun MediaCenter Server FTP Daemon,”** tells you how to use `ftp` to move content between a client machine and a Sun MediaCenter server.

**Chapter 7 “Server Output Data Format,”** describes the format of the data output by the Sun MediaCenter server.

**Chapter 8 “Administering MFS Disks,”** tells you how to monitor disks in the Media File System and how to deal with single and multiple disk failures.

**Chapter 9 “Access Control Lists,”** describes access control lists that you can create and modify for Sun MediaCenter operations.

**Chapter 10 “Troubleshooting,”** presents some problems you might encounter in the use of your server, along with possible solutions.

**Appendix A “Setting up the SNMP Agent,”** tells you how to make use of the SNMP agent that is shipped with the Sun MediaCenter server.

**Appendix B “Creating a Content Package,”** describes how to prepare content if you are unable to use `ftp` for loading content.

The **Glossary** lists terms from the realm of video technology, as well as terms specific to the Sun MediaCenter product.

---

## What Typographic Changes Mean

The following table describes the typographic changes used in this book.

**TABLE P-1** Typographic Conventions

Typeface or Symbol	Meaning	Example
AaBbCc123	The names of commands, files, and directories; on-screen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name%</code> You have mail.
<b>AaBbCc123</b>	What you type, contrasted with on-screen computer output	<code>machine_name%</code> <b>su</b> Password:
<i>AaBbCc123</i>	Command-line placeholder: replace with a real name or value	To delete a file, type <code>rm filename</code> .
<i>AaBbCc123</i>	Book titles, new words or terms, or words to be emphasized	Read Chapter 6 in <i>User's Guide</i> . These are called <i>class</i> options. You <i>must</i> be root to do this.

---

## Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

**TABLE P-2** Shell Prompts

Shell	Prompt
C shell prompt	machine_name%
C shell superuser prompt	machine_name#
Bourne shell and Korn shell prompt	\$
Bourne shell and Korn shell superuser prompt	#

# Introduction

---

<i>Introduction</i>	<i>page 1-1</i>
<i>Sun MediaCenter Hardware</i>	<i>page 1-2</i>
<i>Sun MediaCenter Software</i>	<i>page 1-4</i>
<i>Video Output Interfaces</i>	<i>page 1-8</i>
<i>Content Preparation and Loading</i>	<i>page 1-10</i>
<i>Server Management</i>	<i>page 1-11</i>
<i>Push Model</i>	<i>page 1-11</i>
<i>Design Benefits</i>	<i>page 1-12</i>

---

## 1.1 Introduction

The Sun MediaCenter server is a combination of standard server hardware and special software that is optimized for and dedicated to the storage and delivery of video streams. The special software includes:

- a modified Solaris kernel;
- modified network interface drivers, dedicated to continuous-media output;
- a Media File System (MFS), optimized for the demands of the delivery of isochronous bit streams;
- a Media Stream Manager (MSM), which includes an RPC-based API, that provides users access to the server;

- a Content Manager (CM), which includes an RPC-based API that allows users to move content between servers or between a server and a client.
- Web-based tools that allow you to configure a Sun MediaCenter server, monitor server operations, and transfer video content to a server.

The Sun MediaCenter server delivers MPEG bit streams at a constant bit rate. The content of the streams are stored on an array of disks. The server guarantees that, unless the server hardware fails, once a stream request is accepted, it will be delivered at the specified constant bit rate until the stream ends or the server is told to stop.

Using a Sun MediaCenter server, you can begin playing a stream within seconds after storage of that stream on the server has started. This feature is called “playthrough”.

The Sun MediaCenter server protects against single-disk failures by implementing a parity mechanism similar to RAID Level 4 (Block-Interleaved Parity).

---

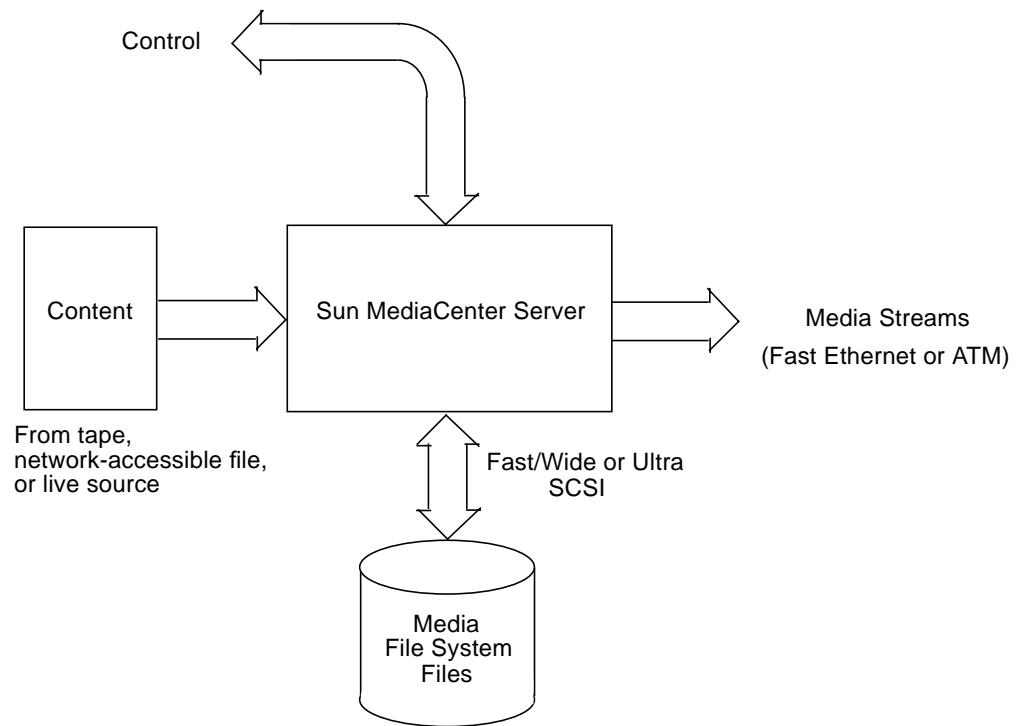
## 1.2 Sun MediaCenter Hardware

The Sun MediaCenter product line is built upon existing Sun hardware and software. This provides you with excellent performance (especially in terms of cost per stream) and flexibility in interfacing Sun MediaCenter servers to their environment.

Each model of the Sun MediaCenter product line is designed to maximize input/output throughput from the server's disk subsystem through a network interface, to an output device. Each model is equipped with a large amount of RAM, for buffering streams and handling online content loading, and a large amount of disk storage.

Within a disk subsystem, specific disk models are chosen for their appropriateness for multimedia applications.

The Sun MediaCenter hardware architecture is shown in FIGURE 1-1.



**FIGURE 1-1** High-level Hardware Architecture

Content is loaded onto the Sun MediaCenter server via the server's Content Manager, over a LAN or using 4- or 8-mm DAT tape. Video streams are controlled through network interfaces distinct from the network interfaces used for video output, using the Media Stream Manager (RPC-based) interface described in the *Sun MediaCenter Server Programmer's Guide*.

## 1.3 Sun MediaCenter Software

The Sun MediaCenter software has three major components, the Media File System (MFS), the Media Stream Manager (MSM), and the Content Manager (CM). (In addition to these, there are special network interface drivers, which are tightly coupled with the MFS.) The relationship of these modules is illustrated in FIGURE 1-2.

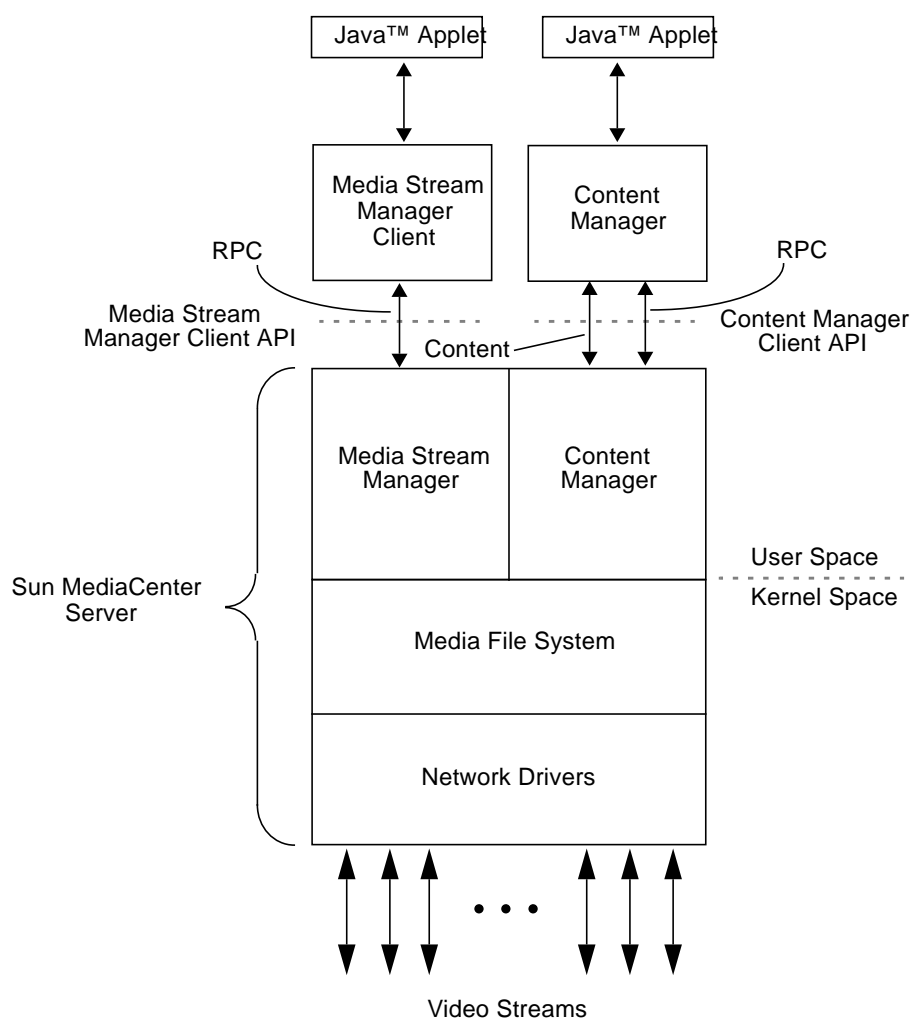


FIGURE 1-2 Sun MediaCenter Software Architecture



With the exception of the APIs for the MSM and CM, the workings of the server's software components are largely invisible to the user. This is particularly true of the MFS, access to which is exclusively throughout the MSM and CM.

The Sun MediaCenter software offers the following features:

*Guaranteed stream delivery rate*

Once a stream is accepted for delivery, the server delivers the stream contents at a guaranteed rate until the end of the stream content is reached, the server is asked to stop playing the stream, or server hardware fails.

*Independent streams*

Delivered streams are independent: they can be stopped and started individually; output streams might all be playing simultaneously from different portions of the same content stream or from different content streams.

*Online loading*

The server maintains delivery of streams as the highest priority task, so that as the server approaches its maximum bandwidth in stream delivery, any content loading taking place will slow and, at some point, stop, until such time as the server recovers sufficient resources to resume loading. (Note that you cannot load content if the server is reading data from the parity disk, which occurs following a disk failure.)

*Playthrough*

The server software supports a playthrough capability wherein a Sun MediaCenter server can deliver streams from a title as that title is stored. A given piece of content can be played from the server only five seconds after storage of it has begun.

The following subsections describe the components of the Sun MediaCenter software.

## 1.3.1 Media File System (MFS)

The MFS is designed to deliver multimedia data from an array of attached disks to an output network. In addition to providing the foundation for all of the features listed above, the file system supports:

*Recovery from single-disk failures*

The MFS will recover from a single disk failure after a small start-up interval (less than 10 seconds). It uses a strategy similar to RAID-4. The number of content disks for each parity disk varies with the disk subsystem used for a particular server model.

### *Hot swappability*

The MFS supports the replacement of a failed disk while the server continues to deliver streams. Following disk replacement, you use a single command to restore data from the parity disk to the new disk and return the server to normal playback mode. The procedure for hot swapping a disk is described in Chapter 8, "Administering MFS Disks."

### *Efficient use of disk bandwidth*

The file system is designed to extract a significant portion of the available disk bandwidth (50-90%). Content is striped across the disk array.

The MFS compensates for thermal recalibration delays.

### *Efficient transfer from disk to output*

The MFS is designed to transfer data from the disk array to the output network with a minimum of processor involvement.

### *Large file and filesystem size*

An MFS file can be as large as  $(2^{64}) - 1$  bytes in length; an MFS can contain up to  $2^{31}$  files. In practice, file size and filesystem size are limited by available disk space rather than any limits imposed by the MFS.

### *High-resolution rate control*

The MFS controls the rate of output so that any one output packet may be delivered within the resolution of the underlying hardware/software system.

### *Random access*

The controller of a stream may stop and start a stream at nearly arbitrary points. The server maps these requests to the nearest content packet (for example, MPEG-2 Transport Packet) boundary. This random access feature is used in trick play. Section 2.2, "MPEG Encoding Requirements for Trick Play," on page 2-2 for a discussion of splice points in trick-play files.

### *Admission control*

The MFS will not accept a request for a new stream that would cause it to fail to deliver all its streams at the request data rate.

### *File locking*

The MFS ensures that streams that are currently being played are not removed.

The MFS consists of five basic subsystems:

- file system layout manager

The MFS layout manager allocates and deallocates disk blocks to content using striping and parity algorithms.

- scheduler

The scheduler sorts and schedules disk I/O, manages bit pump buffers, performs admission control, and schedules output driver activity.

- bit pump

The bit pump is responsible for allocating internal buffering and efficiently transferring data between the disk and the output drivers using the internal buffers. It is responsible for fine-grain scheduling and rate control of the output driver. The bit pump is dependent on the particular output driver used, whether it be for Fast Ethernet or ATM.

- disk driver

The output driver is responsible for low-level output device management and encapsulating outgoing data according to the rules of the output medium, and for efficiently transferring data from the internal buffers to the output medium.

See Chapter 7 for a description of the format of output data.

- output driver

The disk driver is responsible for efficiently transferring disk data to the fixed internal buffers. It consists of a SCSI disk driver and a modified SCSI host adapter driver.

The MFS components listed above are illustrated in FIGURE 1-3.

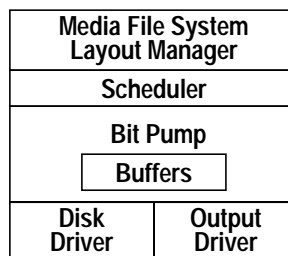


FIGURE 1-3 Media File System Components

## 1.3.2 Media Stream Manager

The MSM controls the video streams stored in the MFS. The MSM implements a set of procedures that interact with the MFS to allow you to stop, start, pause, and resume video bit streams. A central notion of the MSM is the playlist, from which titles can be played on a time-synchronized basis. Each playlist is associated with a player, which can be passed among multiple clients.

The MSM's primary interface is the MSM Client API, described in the *Sun MediaCenter Server Programmer's Guide*.

The MSM supports playthrough, trick play, and other features of the Media File System.

### 1.3.3 Content Manager

The CM provides for the loading of content onto a Sun MediaCenter server. Server requirements for content, such as Table of Contents files and index files (for trick play) are handled transparently by the CM. The CM also supports the movement of content between Sun MediaCenter servers and between a server and a client. This capability provides for the backing up and restoring of video files.

The CM has a server-based set of RPCs and a SPARC client-based library, both described in the *Sun MediaCenter Server Programmer's Guide*. Chapter 4 describes a Web-based tool for moving content onto a Sun MediaCenter server. Chapter 5 presents a set of utilities based on CM. Chapter 6 describes the Sun MediaCenter FTP daemon, which is also based on CM.

### 1.3.4 Web-based Tools

The following Web-based tools are provided with the Sun MediaCenter server software:

- The Configure utility allows you to configure a Sun MediaCenter server. This tool is described in the *Sun MediaCenter Installation and Configuration Manual*.
- WebAdmin allows you to monitor server operations. This tool is described in Chapter 3.
- CmTool allows you to transfer video content to a server. This tool is described in Chapter 4.

---

## 1.4 Video Output Interfaces

Sun MediaCenter software is designed to output streams over ATM or Fast Ethernet, using SunATM™ or SunFastEthernet™, respectively. With both types of output, the streams are delivered at a constant bit rate; there is no dynamic flow control or error recovery protocol. This is referred to as the “push” model of stream delivery and is discussed in Section 1.7, “Push Model,” on page 1-11.

### 1.4.1 ATM

In an ATM environment, one or more Sun MediaCenter servers are directly connected to an ATM switch via ATM host adapters. Digital video streams are sent to clients using ATM virtual circuits.

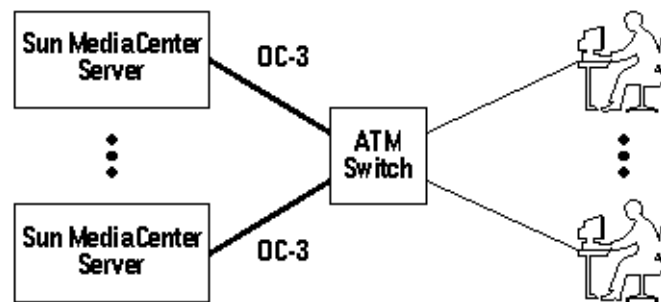


FIGURE 1-4 Sun MediaCenter Servers in ATM Environment

### 1.4.2 LAN

In the LAN environment, each Sun MediaCenter server is connected to one high-speed 100BaseT subnetwork via one or more Fast Ethernet host adapters. Digital video streams are sent to their destination(s) using LAN packets. Destination clients can access the video streams by connecting directly to one of the 100BaseT networks. Alternatively, the 100BaseT networks can be connected to an optional packet-filtering switch that routes to independent 10BaseT networks. This allows individual destination clients to connect to a 10BaseT network without affecting the aggregate bandwidth available.

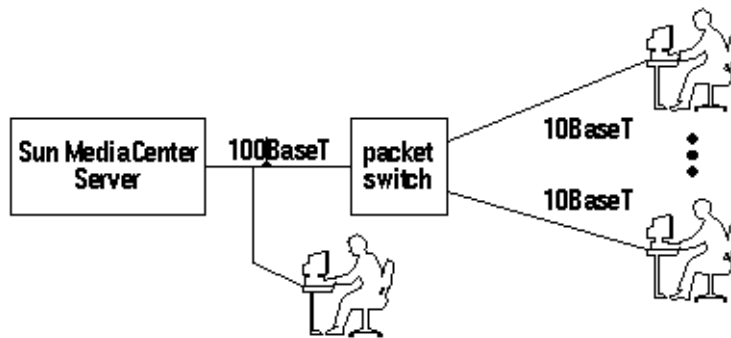


FIGURE 1-5 Sun MediaCenter Servers in LAN environment

---

## 1.5 Content Preparation and Loading

The Sun MediaCenter software specifies a format for content, described in Chapter 2, “Preparing and Loading Content.”

Sun MediaCenter software uses the Content Manager for storing video content on the server. The Content Manager offers the following interfaces:

- A set of RPCs that enable moving of content between local storage and a Sun MediaCenter server. Also, there are RPCs to delete and rename content and to manipulate access control lists.
- A client library based on the RPCs described in the preceding bullet. This library provides you the features of the RPCs and frees you from having to deal with network programming.
- CmTool, a Web-based interface for transferring video content onto a Sun MediaCenter server.
- A set of utilities based on the Content Manager. These include `smc_tar`, `smc_copy`, and a Sun MediaCenter FTP daemon. In some cases, these utilities automatically create the administrative files that support a bit stream.

Once content is stored in the MFS, you use the MSM to play it. The MSM supports VCR-like functions, such as play, pause, fast-forward (at multiple speeds), and reverse.

---

## 1.6 Server Management

An SNMP agent is provided with the Sun MediaCenter software. This agent includes the following MIB information:

- current number of playing streams;
- current aggregate bandwidth;
- number of missed deadlines;
- number and location of disk errors.

The SNMP agent also supports the following traps:

- Missed\_Deadline
- Admission\_Failed
- Disk\_Failure

See Appendix A for a description of and instructions for configuring the SNMP agent.

---

## 1.7 Push Model

The Sun MediaCenter server implements the “push” model of bit stream delivery. In this model, the server initiates and delivers bit streams over a network interface dedicated to video output. On this interface, there is no “back channel” for decoder-to-server flow control. The burden of staying within the time constraints of bit streams is entirely on the server. On the receiving end, the decoder must be fast enough to keep up with the server, but does not bear any responsibility for maintaining timely delivery.

The push model differs from the “pull” model, in which the receiving device engages a video server in a protocol for connection setup and bit-stream delivery. Both server and receiver must be able to look inside the bit stream to interpret and, possibly, take action, based on the contents of MPEG packets.

The push model has an advantage over the pull model in that receiving devices do not require intelligence for communicating with the server. Receiving devices can be simpler and less expensive than pull-model receivers.

Consistent with the push model, the Sun MediaCenter server’s supporting software, including the MSM, reflects the one-way communication between the server and receiving devices. For example, network interface drivers used for bit stream

delivery support output only. The MSM Client API has calls that allow you to specify an ultimate destination, but do not, for example, allow you to set any parameters on that destination, which would imply the ability of the receiver to confirm a setting.

---

## 1.8 Design Benefits

The Sun MediaCenter server is more than simply a standard server with large amounts of memory and disk space. It is the Sun MediaCenter software—the MFS in combination with network drivers optimized for video output—that distinguishes the Sun MediaCenter server from its standard counterpart. This software provides the following advantages over a standard server:

- A Sun MediaCenter server can guarantee a certain specified level of stream delivery, with defined limits to jitter and drift. This means that, unless server hardware fails, a Sun MediaCenter server will continue to deliver video bit streams, up to its maximum output bandwidth, at the rate specified in those streams.
- Because the Sun MediaCenter server is dedicated to and optimized for video-bit-stream delivery, the server can support more bit streams for the same amount of hardware resources (memory, disk space, and network interfaces), as compared to a standard server. This means that a Sun MediaCenter server has a lower cost per stream than a comparably equipped standard server that is used as a video server.
- Because the MFS is tailored to high I/O bandwidth applications, the Sun MediaCenter server can guarantee average-case disk I/O. Servers that rely on standard file systems can guarantee only worst-case disk I/O. The MFS ensures that whether users request different bit streams, up to maximum bandwidth, or use up the maximum bandwidth with requests for the same bit stream, the server's guarantee of timely delivery is not affected. With a file system not optimized for video, certain user requests, such as multiple requests for the same bit stream, are disruptive of server performance.



## Preparing and Loading Content

---

<i>General MPEG Encoding Requirements</i>	<i>page 2-1</i>
<i>MPEG Encoding Requirements for Trick Play</i>	<i>page 2-2</i>
<i>Splice Points</i>	<i>page 2-3</i>
<i>Content Loading</i>	<i>page 2-5</i>
<i>References</i>	<i>page 2-7</i>

This chapter specifies what video content should consist of and how it should be formatted, as required by the Sun MediaCenter software. It also provides guidelines for content loading.

This chapter does not tell you how to encode content. The Sun MediaCenter server has successfully stored and delivered content encoded on a variety of different encoders, from a number of different vendors.

---

### 2.1 General MPEG Encoding Requirements

This section defines general MPEG encoding requirements. A subsequent section defines additional constraints to the general requirements.

To play correctly on the Sun MediaCenter server, a bit stream must meet *all* of the following requirements:

- All bit streams in a content package must be encoded as either (1) MPEG-2 Transport Streams or Program Streams, conforming to the ISO/IEC 13818-1 standard, or (2) MPEG-1 System Streams, conforming to the ISO/IEC 11172-1 standard.

- The timing information embedded in an MPEG file must exactly reflect the constant bit rate encoded in the file. Specifically, the time differences between any two consecutive PCR (SCR for Program Streams or MPEG-1 System Streams) time stamps must exactly reflect the time required to play out the number of bytes between the two PCRs (SCRs for Program Streams or MPEG-1 System Streams) at the bit stream's specified bit rate.
- The bit stream must be encoded at constant bit rate, between 1.5 and 15 Mbit/sec. You can encode a bit stream at any rate within this range.
- Each bit stream must contain one and only one Program Association Table (PAT), but these can be repeated as often as required. Multiple programs in one transport stream is not supported.
- Each bit stream must contain one and only one video stream.
- For MPEG-2 Transport Streams, each MPEG-2 transport packet that contains the Program Clock Reference (PCR) must occur in an even-numbered transport packet, where the first transport packet in the stream is defined to be numbered 1, and is odd. This is necessary to meet the ATM Forum MPEG-over-AAL5 requirement that, in the case where there are two MPEG-2 transport packets per AAL5 packet, the PCR must be in the last transport packet in the AAL5 packet.

---

## 2.2 MPEG Encoding Requirements for Trick Play

Trick play refers to the manipulation of video bit streams to create the effect of fast forward and rewind, as found in conventional VCRs. To perform trick play, the Sun MediaCenter server does not dynamically manipulate data within the normal-play bit stream. Instead, the server switches from a normal-play stream to the appropriate position in another bit stream that has the requested play speed and/or direction.

To enable trick play on the Sun MediaCenter server, you must provide a set of distinct MPEG bit streams, beyond the bit stream that represents the movie at the normal playback speed. These additional bit streams represent the content at several speeds, in either the forward or reverse direction. If a content package does not contain at least one additional bit stream beyond the bit stream for normal playback, trick play functions are disabled.

All trick files must meet the following requirements:

- All trick files must contain the identical video and audio PIDs as those of the file at normal play speed.
- All trick files must be encoded in a format (that is, MPEGTS, MPEGPS, or MPEG1SYS) identical to the bit stream at normal play speed.

- All trick files must be encoded at a bit rate that is less than or equal to the bit rate of the normal-speed bit stream.

The Media Stream Manager supports the playing of trick play streams. See the chapter on the Media Stream Manager Client API in the *Sun MediaCenter Server Programmer's Guide* for a discussion of how to play titles that have trick play streams.

---

## 2.3 Splice Points

To facilitate switching among multiple bit streams within a content package, all bit streams must define convenient splice points. A splice point is a position in the bit stream at which the server switches into or out of when a user switches from one speed or direction to another. For optimum switching among streams, you should specify splice point locations in an index file, the format for which is described in Section B.5 “Index File Requirements” on page B-10.

---

**Note** – The Media Stream Manager switches among trick play streams and a normal play stream even in the absence of an index file. However, if an index file is not present, the Media Stream Manager performs positioning between streams by the linear interpolation of bit rate over file size, which means that streams are entered and exited at arbitrary points. This sort of stream switching causes problems for some decoders. You should use trick play without an index file only if your decoder is capable of resynchronizing and if artifacts are acceptable.

---

The frequency of the splice points within the bit streams in a content package should reflect the latency you want for trick play functionality. More splice points mean lower latency.

A splice point must have the characteristics described in the following subsections.

### 2.3.1 Packet Boundary

A splice point must occur on the boundary of the underlying packet. For example, for the MPEGTS format, the splice point should occur on the MPEG-2 transport packet boundary.

### 2.3.2 closed\_gop

The closed\_gop is a one-bit flag that indicates the nature of the predictions used in the B-pictures immediately following the first coded I-frame following the group-of-pictures header. The closed\_gop must be set to 1 to indicate that these B-pictures have been encoded using only backward prediction.

### 2.3.3 payload\_unit\_start\_indicator

For an MPEG-2 Transport Stream, the first MPEG-2 transport packet following a splice point must have the payload\_unit\_start\_indicator set, and the packet must contain a payload.

### 2.3.4 adaptation\_field\_control

For an MPEG-2 Transport Stream, the first MPEG-2 transport packet following a splice point must have the adaptation\_field\_control set to indicate an adaptation field as well as payload (value of 11).

### 2.3.5 random\_access\_indicator

For an MPEG-2 Transport Stream, the first MPEG-2 transport packet following a splice point must have the random\_access\_indicator set to 1.

### 2.3.6 PCR\_flag

For an MPEG-2 Transport Stream, the first MPEG-2 transport packet following a splice point must have the PCR\_flag set to 1, and must contain the PCR value in the adaptation field.

---

**Note –** The above requirements for splice points apply to the video stream. If the bit stream contains audio, and/or other elementary streams, the requirements apply to those streams as well.

---

---

## 2.4 Content Loading

The preferred method of storing content on a Sun MediaCenter server is to use the Web-based CmTool. CmTool uses the Sun MediaCenter FTP daemon and the Content Manager (CM), part of the Sun MediaCenter software, to store content. The CM automatically extracts from MPEG content the information the server needs to successfully store and play that content. See Chapter 4 for information on using the CmTool interface.

The Sun MediaCenter software also offers the following alternatives for transferring video files:

`smc_tar`

Can copy to or extract from a tape device. Takes a content package as input for content loading. Copies between a `tar` device or a Unix File System (UFS) and the MFS.

`smc_copy`

Takes a title name as input. Use from server to server. Can create a TOC file and index files if none present. Copies between MFSs.

`ftp` in conjunction with the Sun MediaCenter FTP daemon

Requires only the `smc:` keyword. Works on any platform on which an `ftp` client can run. Creates a TOC file and index files if none present. Copies between a local file system and the MFS.

`smc_tar` and `smc_copy` are described in Chapter 5. See Chapter 6 for instructions on the use of `ftp` with the Sun MediaCenter FTP daemon.

A file-transfer alternative beyond those listed here is to write a program to the Content Manager (CM) API or CM Client API. These APIs are described in the *Sun MediaCenter Server Programmer's Guide*.

The Sun MediaCenter FTP daemon works with the most commonly-used MPEG formats—MPEG-1 System Stream (MPEG1SYS) and MPEG-2 Transport Stream (MPEGTS).

The following are circumstances in which you *cannot* use `ftp` to store content:

- Your MPEG file is larger than 2.1 GB. Use `smc_tar` instead. With such large files, you must create the correct `segment` clauses in the TOC file. See the description of the TOC file in Appendix B. The 2.1 GB file size is a limitation of Solaris 2.4, not the Sun MediaCenter server.
- Your content is in a format other than MPEG1SYS or MPEGTS. If your stream is not encoded in one of these formats, you must use `smc_tar` to load content.

- The machine on which your content is stored does not have `ftp` client software.
- For video file transfers from the server to a client machine, if your video file is smaller or equal to 2.1 GB, use `smc_copy` or `ftp`. If the file is larger than 2.1 GB, use `smc_tar`.
- For video file transfers between Sun MediaCenter servers, use `Cmtool` or `smc_copy`.
- If you want to perform a “third-party” transfer, in which you invoke a command on one machine to transfer video files between a second and a third machine, use `smc_copy`.
- To copy to or extract from a tape device, use `smc_tar`.
- Both `ftp` and `smc_tar` are useful for backup and restore of video files.

If any of these circumstances applies to you, you must create a content package as described in Appendix B, then use `smc_tar` to load your content. See Chapter 5 for a description of `smc_tar`.

When you copy content to a Sun MediaCenter server (which includes the narrower case of content loading), you, the copier, have certain access to that title that is not available to other users. You use the `smc_settacl` utility to change access to the copied title. See Section 5.1 “Setting a Title Access Control List” on page 5-1 an explanation of setting permissions for copied titles.

A couple of points on content loading:

- A Sun MediaCenter server can play streams while you are loading content.
- You can begin to play a title seconds after you have started loading it onto the server. This feature is called “playthrough” and is discussed in detail in the *Sun MediaCenter Server Programmer’s Guide*.
- For video formatted as MPEG-2 Transport Streams, Sun MediaCenter software automatically generates index files for titles that contain trick play streams.

When storing content, free space in the UFS `/var` partition might be a consideration. While the bulk of content storage is handled by the disk subsystem controlled by the MFS, the MFS software uses space in `/var` for keeping track of MFS files. Specifically, the MFS uses:

- a small amount of space for TOC and index files (these are described in Appendix B);
- a fixed amount, 1500 bytes, for each MFS file;
- one byte for every 10 minutes of play time in a title.

The last requirement is a function of play time, not encoding rate or MFS file length.

As examples of MFS space consumption in `/var`, consider that a two-hour MPEG-1 title consumes 73.5 KB; a 30-second title consumes 1800 bytes. Our testing has shown free space of 10 MB on a Sun MediaCenter UltraSPARC to be adequate for a set of titles that places an extremely high demand on `/var` space.

---

## 2.5 References

ISO/IEC 13818-1/2/3 Recommendation H.262: “Generic Coding of Moving Pictures and Associated Audio Information: Systems/Video/Audio”

ISO/IEC 11172-1/2/3 “Coding of Moving Pictures and Associated Audio for digital storage media at up to 1.5 Mbit/s: Systems/Video/Audio”





## Monitoring the Server

---

<i>Overview</i>	page 3-1
<i>Using the WebAdmin Tool</i>	page 3-2
<i>Resource Utilization</i>	page 3-5
<i>Streaming Videos</i>	page 3-6
<i>Copy Sessions</i>	page 3-9
<i>Disk Capacity</i>	page 3-12
<i>Troubleshooting</i>	page 3-15

This chapter describes the Sun MediaCenter WebAdmin tool, a Web-based graphical user interface (GUI) for monitoring the Sun MediaCenter server. The tool is part of the software that is installed on the Sun MediaCenter server.

---

### 3.1 Overview

The Sun MediaCenter WebAdmin tool is a set of Java™ applets that allow you to monitor the following:

- Viewing of streams on a Sun MediaCenter server.
- Movement of content to or from a Sun MediaCenter server.
- MFS disks.

---

**Note** – Any user can use the WebAdmin tool to view server status. However, for certain actions, such as deleting a stream or aborting a copy session, you will be prompted for a Sun MediaCenter server username and password.

---

---

## 3.2 Using the WebAdmin Tool

You can run WebAdmin from the following:

- Any browser that is fully compatible with Java Developer's Kit (JDK) version 1.1.1 or later (for example, HotJava™ 1.0 or later).
- Java Applet Viewer available with JDK 1.1.3.

The server that you want to monitor must have the Sun MediaCenter `SUNWsmcswa` package installed; this software is available on the current release of the Sun MediaCenter server. The server must also be a Web server.

### 3.2.1 Invoking the Tool

To invoke the WebAdmin tool, enter the following URL in your browser window:

`http://<servername>/smc.html`

...where `<servername>` is the name of the Sun MediaCenter server that you want to monitor. If you have more than one server that you will be monitoring through the WebAdmin tool, you can simply re-enter the URL with the appropriate server name. At the Sun MediaCenter page, shown in FIGURE 3-1, select WebAdmin.



FIGURE 3-1 Sun MediaCenter Page

You may need to set security on your browser to allow the WebAdmin applet to connect to your local system. For the Sun HotJava browser, select "Allow all connects without warning" and "Allow access to all properties" in the Security Violation popup windows.

Once the applet loads, the Resource Utilization page is displayed:

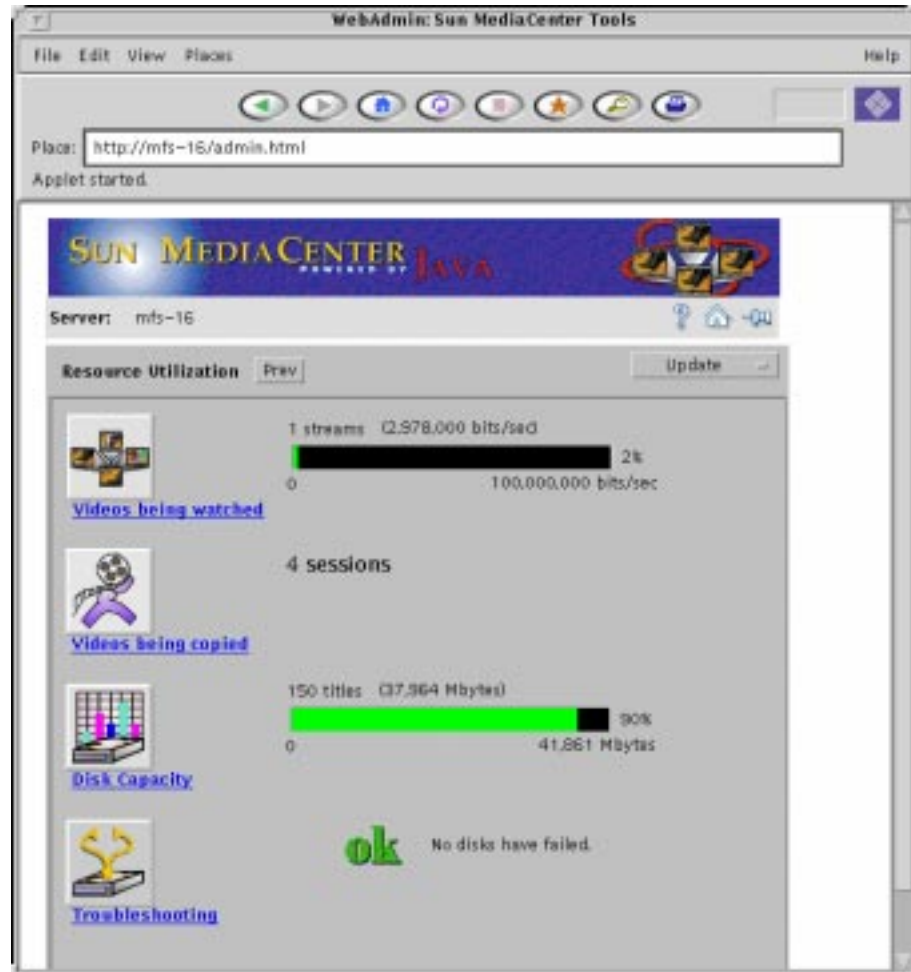


FIGURE 3-2 Resource Utilization Page

## 3.2.2 Common Page Elements

This section describes the items that appear at the top of each page in the WebAdmin tool. Each page is described in detail in later sections in this chapter.

The Server Name field appears at the top left corner of each page and identifies the Sun MediaCenter server being monitored. The following icons appear at the top right of the pages:

- The question mark displays online help for the WebAdmin tool.
- The house icon displays the “home” page, which is the Sun MediaCenter page.
- The pin icon, when selected, keeps the current page displayed in the workspace, even if you go to a different page. For example, you can “pin” the Resource Utilization page in your workspace while you are viewing information on a particular content movement session.

The title of the WebAdmin page appears below the Server Name. The Next or Prev buttons that appear next to the page title are useful for traversing the pages in the tool.

By default, data in each WebAdmin page is updated every 10 seconds. The Update menu allows you to select the “Update Now” option to get the most current data. Or, you can select the Timed Update option to display a dialog box where you can change the interval at which data in the pages is updated.

The remaining sections in this chapter describe each of the pages in the WebAdmin tool.

---

## 3.3 Resource Utilization

The Resource Utilization page, shown in FIGURE 3-2, displays an overall view of Sun MediaCenter server operations. It contains a graphic and a link to a more detailed page for each of the following items:

### **Videos being watched**

Displays the streams that are being played on the server and the users who are playing the streams. The graph shows the number of streams that are being played and the percentage of the available bandwidth that the streams are using. Click on the icon or the link to display the Streaming Videos page, where you can view more information about the streams. See Section 3.4, “Streaming Videos,” on page 3-6.

**Videos being copied**

Displays the number of content movement sessions on the server. Click on the icon or the link to display the Copy Sessions page, where you can view more information about the content movement sessions. See Section 3.5, “Copy Sessions,” on page 3-9.

**Disk Capacity**

Displays the status of the MFS disks. The graph shows the number of titles stored in the MFS and the percentage of available disk space being used. Click on the icon or the link to display the Disk Capacity page, where you can view more information about the MFS disks. See Section 3.6, “Disk Capacity,” on page 3-12.

**Troubleshooting**

Allows you to monitor the following components: MFS disks, network, console, and bit pump. The overall status of the disks is indicated by the graphic; if the graphic is green, then all disks on the server are operating normally. If the graphic is red, one or more disks on the server need to be checked for physical malfunction. See Section 3.7, “Troubleshooting,” on page 3-15.

When you click on the icon for one of the above items, the page for that item is displayed. Each of these pages is explained in this chapter.

---

## 3.4 Streaming Videos

The Streaming Videos page shows the streams that are being played on the selected Sun MediaCenter server and the users that are playing the streams.

The Streaming Videos page is shown below.

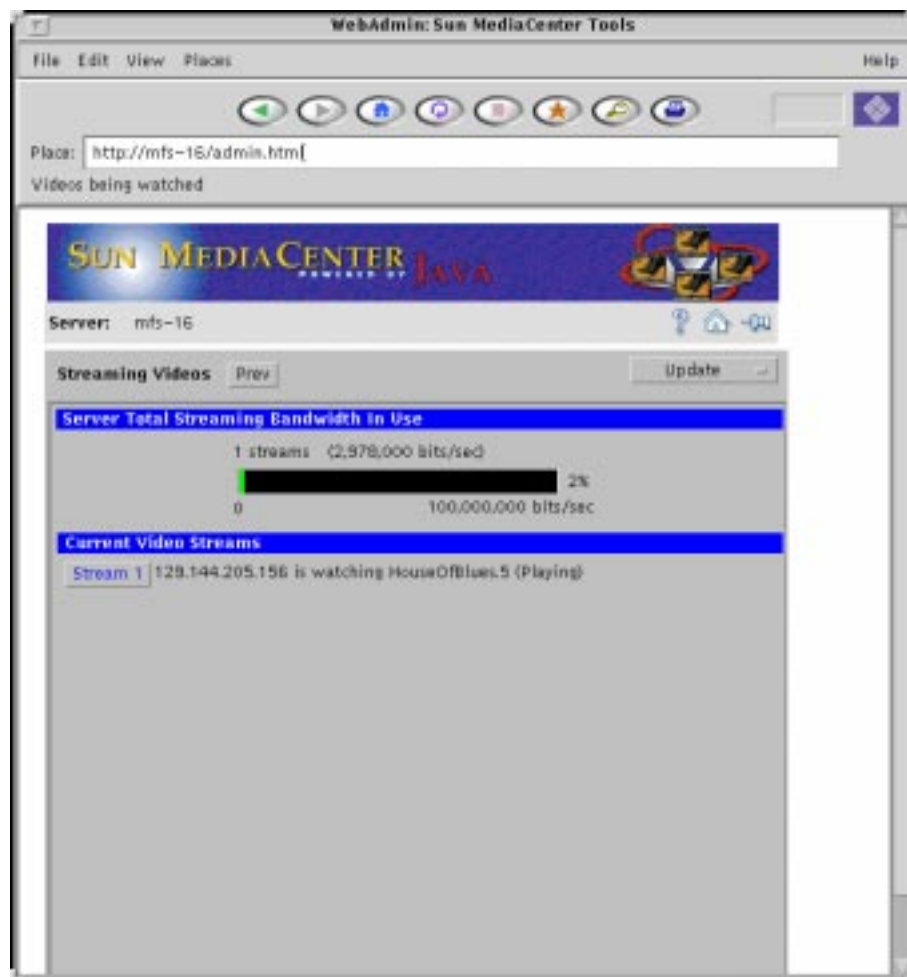


FIGURE 3-3 Streaming Videos Page

The top portion of the Streaming Videos page shows the number of streams that are being played on the server, the average rate of the streams (in bits per second), and the percentage of available bandwidth being used by the streams.

The lower portion of the Streaming Videos page shows the following information about each stream that is being played on the server:

- the TCP/IP hostname or the ATM port number of the user who is viewing the stream

- the title being viewed
- speed (Fast-Forward, Slow Motion, Stopped, Playing, Reverse)

Click on the stream icon to view more information about this stream. The Streaming Videos page shows detailed information about the stream that you selected in the Streams page; this is shown in FIGURE 3-4.



FIGURE 3-4 Streaming Videos Page with Stream Detail



This page shows the following information about the stream being viewed:

**Stream Name and Direction**

Title and direction (fast-forward, playing, stopped, reverse, etc.) of the stream.

**Format**

Format of the content.

**Bit Rate**

Bit rate at which the stream is being played.

**Destination**

TCP/IP port or ATM virtual circuit.

**Playlist**

If this stream is part of a playlist, the titles in the playlist and the total time of the playlist are shown. If there are multiple titles in the playlist, an arrow (->) is shown next to the one being played. If a title is being played continuously, an additional icon appears next to the title.

The Delete this stream button allows you to stop the playing of the stream. You must have the proper permissions to do this, as you will be prompted for your username and password.

---

## 3.5 Copy Sessions

The Copy Sessions page shows the status of content movement sessions on the server.

The Copy Sessions page is shown below:



FIGURE 3-5 Copy Sessions Page

The top of the Copy Sessions page shows the number of copy sessions that are active on the server.

The lower portion of the Copy Sessions page shows the following information about each session on the server:

- title of the content being transferred
- speed of the content being transferred (1000 is normal play speed)
- status of the CM session ("being created" or "Opened")

Click on the session icon to view more information about this stream. The Session Information page shows detailed information about the content movement session that you selected in the Copy Sessions page.

The Session Information page is shown below:



FIGURE 3-6 Session Information Page

This page displays the following information:

State

Creating or Opening  
Shared or Exclusive  
Reliable or Unreliable

Status

OK or error message from CM\_API.

Transfer Mode

TCP or ATM

Address

TCP port or ATM address.

Bytes Xfered

The number of bytes already transferred.

Bytes total

The total number of bytes to be transferred.

The bar at the bottom of page indicates the percentage of bytes transferred.

The Abort This Session button allows you to stop the content transfer. You must have the proper permission to do this, as you will be prompted for your username and password.

---

## 3.6 Disk Capacity

The Disk Capacity page shows the status of individual disks on the Sun MediaCenter server.

The Disk Capacity page is shown below:



FIGURE 3-7 Disk Capacity Page

A red, Failed Disk icon appears at the top of this page only if there is a failed disk in the MFS.

The top portion of the Disk Capacity page shows the number of titles that are stored on the server, the amount of disk space used to store the titles, and the percentage of available disk space in the MFS being used by the titles.

The bottom portion of the Disk Capacity page displays the disks in groups, as specified by the RAID layout. Individual disks can be either Data or Parity type. The size and name of each disk are shown.

Click on the disk name to view more information about the disk. The Disk Info page shows detailed information about the disk that you selected in the Disk Information page.

The Disk Info page is shown below:



FIGURE 3-8 Disk Info Page

The top portion of the Disk Info page shows statistics about the usage of the disk.

The bottom portion of the page shows any late events or disk errors.

---

## 3.7 Troubleshooting

The Troubleshooting page allows you to monitor the following server components: MFS disks, network, console, and bit pump.

The Troubleshooting page is shown below:



FIGURE 3-9 Troubleshooting Page

The Troubleshooting page contains the following links:

- **Media File System: disks** displays the Disk Health page. See Section 3.7.1, "Disk Health," on page 3-17.
- **Console Output** displays Sun MediaCenter server console messages. See Section 3.7.2, "Console Log," on page 3-18.
- **Bit Pump Statistics** displays Sun MediaCenter server bit pump information. See Section 3.7.3, "Scheduler and Bitpump," on page 3-20.



### 3.7.1 Disk Health

This page shows the status of disks in the MFS.

The Disk Health page is shown below:

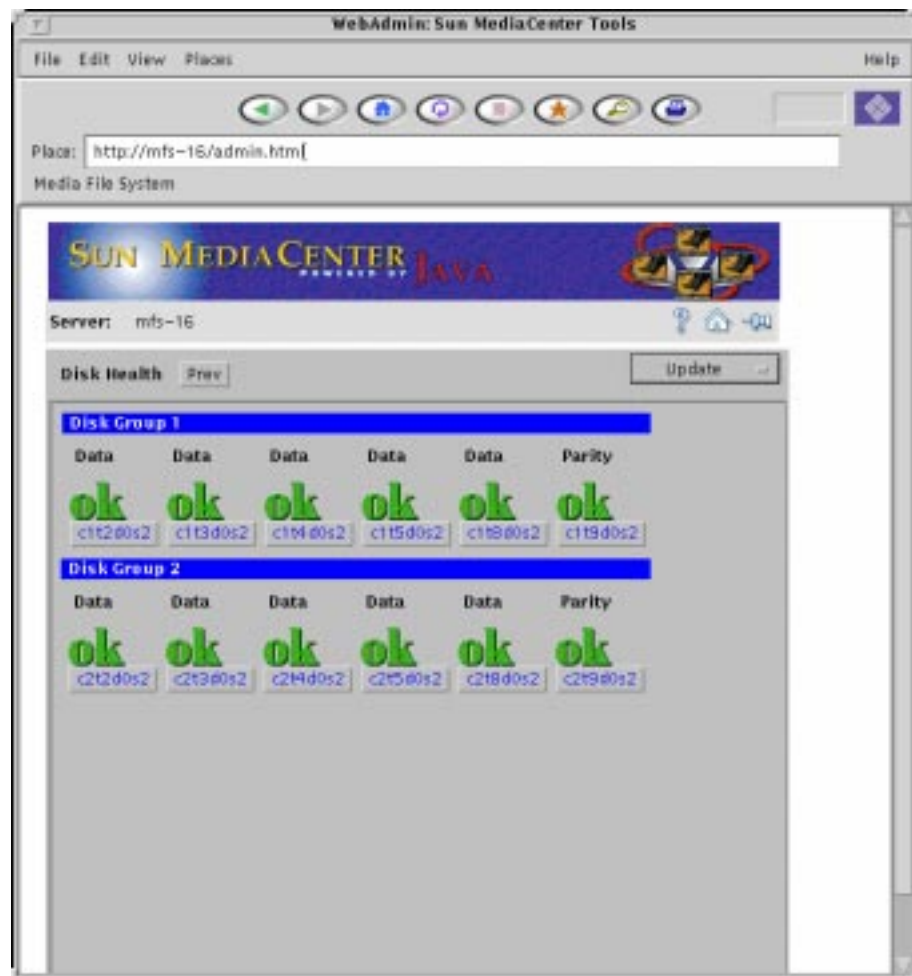


FIGURE 3-10 Disk Health Page

The Disk Health page displays the disks in groups, as specified by the RAID layout. Individual disks can be either Data or Parity type. The name of each disk is shown, along with an icon that shows the status of the disk. Icons can be one of the following colors:

- green is normal
- red is failed or reconstructing

Click on the disk name to view the Disk Info page for more information about the disk. See Section 3.7.1, “Disk Health” for information on the Disk Info page.

## 3.7.2 Console Log

The Console Log page displays the Sun MediaCenter server console messages.

The Console Log page is shown below.



FIGURE 3-11 Console Log Page

You can use the search function at the bottom of the page to search for a specific string. Enter in the string you want to find, then select the Search! button.

### 3.7.3 Scheduler and Bitpump

The Bitpump page displays information about the Sun MediaCenter server scheduler and bitpump.

The Bitpump page is shown below:



FIGURE 3-12 Scheduler and Bitpump Page

TABLE 3-1 describes the fields in this page.

**TABLE 3-1** Scheduler and Bitpump Statistics

Active	
Active Streams	Number of active streams
Bandwidth	Collective network bandwidth of all active streams, in bytes/sec.
System	
Disk Bw	Bandwidth of each disk (without considering seek overhead).
Total Bw	Total bandwidth of all data disks.
Number of disks	Number of data disks.
Number of parity disks	Number of parity disks.
Number of zones	Number of zones per disk. A zone is a region of a disk used for the storage of streams
Period Used	A fixed interval used by VOD scheduler for MFS operations.
Buffer Size	Size of each buffer.
Buffer Maxuse	Maximum number of buffers used at one time.
Buffer Unavail	Number of times buffers not available.
Buffer InUse	A snapshot of the number of busy buffers.
Scheduler	Time period cycles completed.
Online	
Total Bw	Total usable bandwidth for online loading.
Max Transfer	Overall maximum transfer size of online I/O.
Avail Bw	For a given instant, currently available bandwidth per time unit (see bitPump:bitPumpTimePeriod).
Avail Max	Maximum, currently available transfer size for online I/O.
Num Started	Number of online loading requests started.
Num Finished	Number of online loading requests finished.
Kmem Buf in use	Number of buffers in use.
Vod Max Buffers	Maximum number of buffers available.
Vod Buffers in use	Number of buffers in use.

**TABLE 3-1** Scheduler and Bitpump Statistics (*Continued*)

Admit	
Max Active	Maximum number of streams allowed. This a fixed value.
Max Bw	Maximum bandwidth allowed for streams, in bytes/sec.
Num Admitted	Number of successful admissions.
Num Failed	Number of failed admissions.
Total Delay	Total delay for the admission of all streams since last reboot, in milliseconds.
Average Delay	Average admission delay, in milliseconds.
Fail	
Num failed disks	Total number of failed disks.
Recovering	Number of disks that have failed and are undergoing recovery through the parity mechanism. Should be no more than 1.
Latest Fail	Name of the last failed device.
When	Time of the last failed device.
Num Parity Recovery	Number of parity recoveries (XOR).
Late	
Num Late	Number of late disk events.
ID	Instance number of the late event.
Device	Name of the late device.
When	Time at which the late event was detected, in microseconds.
Length	Duration of lateness, in milliseconds.
Errors	
Num Errors	Number of disk errors.
ID	Number identifying the error.
Device	Name of the disk having the error.
When	Time the error occurred.
Error #	Description of the error as returned by the operating system.
Block #	Block number of the errant block.
Size	Number of errant blocks.

**TABLE 3-1** Scheduler and Bitpump Statistics (*Continued*)

bitPump	
bitPumpStreams	Current number of bit pump streams.
bitPumpBandwidth	Bandwidth of the bit pump, in bits/sec. This is a fixed value.
bitPumpMissedDeadlines	Number of missed deadlines.





## Moving Content

---

<i>Overview</i>	page 4-1
<i>Using CmTool</i>	page 4-2
<i>Loading Content</i>	page 4-5
<i>Table of Contents Page</i>	page 4-12

This chapter describes the Sun MediaCenter CmTool, a Web-based graphical user interface (GUI) that allows you to transfer content to a Sun MediaCenter server. The tool is part of the software that is installed on the Sun MediaCenter server.

---

### 4.1 Overview

CmTool is a set of Java applets that allow you to:

- Display the titles that are stored on a Sun MediaCenter server, as well as view information about individual titles.
- Transfer content to a Sun MediaCenter server.

---

**Note** – Chapter 2, “Preparing and Loading Content” describes video content and how it should be formatted for use with the Sun MediaCenter software. You should read this chapter before transferring content to the Sun MediaCenter server.

---

You need to have a valid username and password on the Sun MediaCenter server in order to use CmTool to transfer content to that server. CmTool will prompt you for a username and password when you attempt to load content to the server.

---

## 4.2 Using CmTool

You can run CmTool from the following:

- Any browser that is fully compatible with Java Developer's Kit (JDK) version 1.1.1 or later (for example, HotJava 1.0 or later).
- Java Applet Viewer available with JDK 1.1.3.

The server that you want to move content to must have the Sun MediaCenter `SUNWsmcswa` package installed; this software is available on the current release of the Sun MediaCenter server. The server must also be a Web server.

### 4.2.1 Invoking the Tool

To invoke CmTool, enter the following URL in your browser window:

`http://<servername>/smc.html`

...where `<servername>` is the name of the Sun MediaCenter server that you want to monitor. If you have more than one server to which you will be transferring content with CmTool, you can simply re-enter the URL with the appropriate server name. At the Sun MediaCenter page, shown in FIGURE 4-1, select Content Manager.



FIGURE 4-1 Sun MediaCenter Page

You may need to set security on your browser to allow the CmTool applet to connect to your local system. For the Sun HotJava browser, select "Allow all connects without warning" and "Allow access to all properties" in the Security Violation popup windows.

Once the applet loads, the Table of Contents page is displayed, with a list of content that is currently stored on the server.



FIGURE 4-2 Table of Contents Page

## 4.2.2 Common Page Elements

This section describes the items that appear at the top of each page in CmTool. Specific elements of each page are described later in this chapter.

The Server Name field appears at the top left corner of each page and identifies the Sun MediaCenter server you are managing. The following icons appear at the top right of the pages:

- The question mark displays online help for CmTool.
- The house icon displays the “home” page, which is the Sun MediaCenter page.
- The pin icon, when selected, keeps the current page displayed in the workspace, even if you go to a different page. For example, you can “pin” the Table of Contents page in your workspace while you are viewing information on a particular content movement session.

The title of the CmTool page appears below the Server Name. The Next or Prev buttons that appear next to the page title are useful for traversing the pages in the tool.

By default, data in each page is updated only when the page is displayed. The Update menu allows you to select the “Update Now” option to get the most current data. Or, you can select the Timed Update option to display a dialog box where you can specify that the data in the pages be automatically updated at regular intervals.

The remaining sections in this chapter describe how to use CmTool.

---

## 4.3 Loading Content

The Table of Contents page, shown in FIGURE 4-2, displays a list of the titles found on the Sun MediaCenter server.

To load content onto the Sun MediaCenter server, click on the Load new Content button. This displays the Login Information page, shown in FIGURE 4-3.

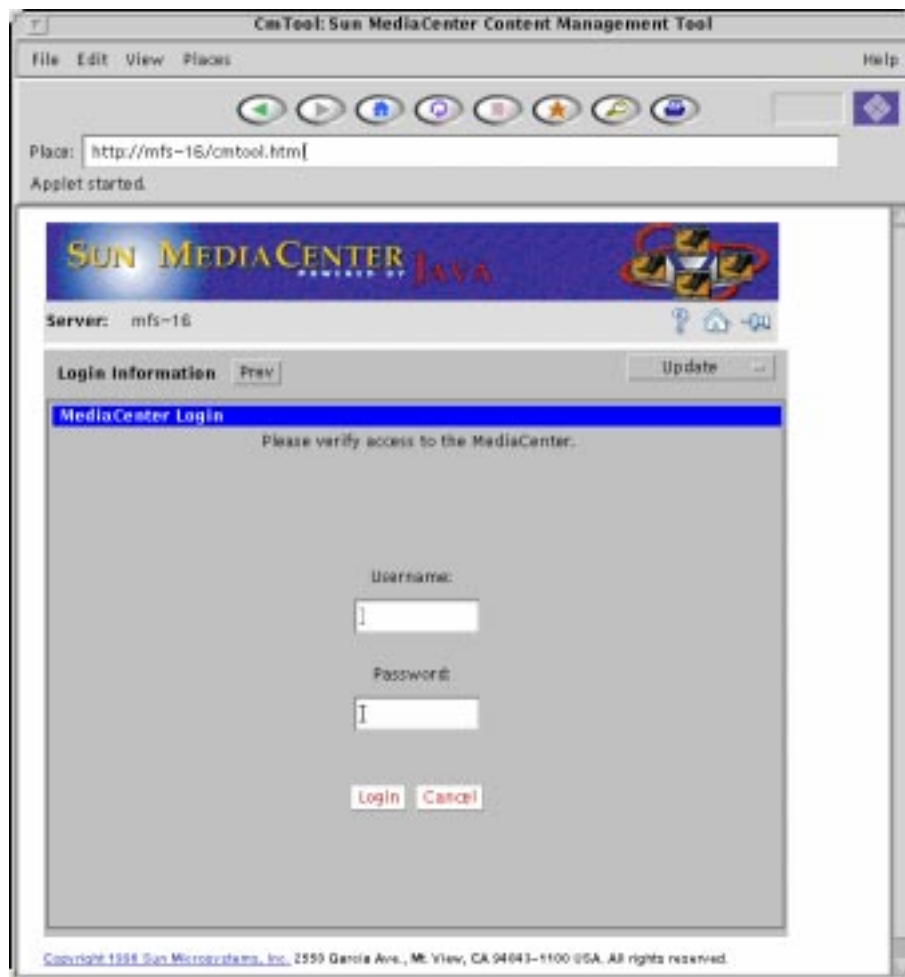


FIGURE 4-3 Server Login Page

You enter in a username and password for the Sun MediaCenter server in the Login Information page. Once you have successfully logged in to the server, you will not need to enter the login information again unless you quit CmTool or reload the page.

Upon successful login, the Load Content page is displayed.



FIGURE 4-4 Load Content Page

The disk space available on the server is shown in the upper right corner of the page. To load content onto the server, first specify the source where the content is located.

If you select Local filesystem, you are prompted for the username and password to log in to the local system.

If you select Remote Machine, you are prompted first for the name of the remote system, then for the username and password to log in to the remote system.

---

**Note** – If you are transferring content from a remote system, you must ensure that the CmTool applet can communicate with other systems on the network. For Sun HotJava, select Preferences from the File menu and select “Medium Security” in the Applet Security box. For Java Applet Viewer, set the Applet Properties section of the Network Access window to “Unrestricted.”

---

In FIGURE 4-5, the selected content source is a remote system; the content on that system is displayed.

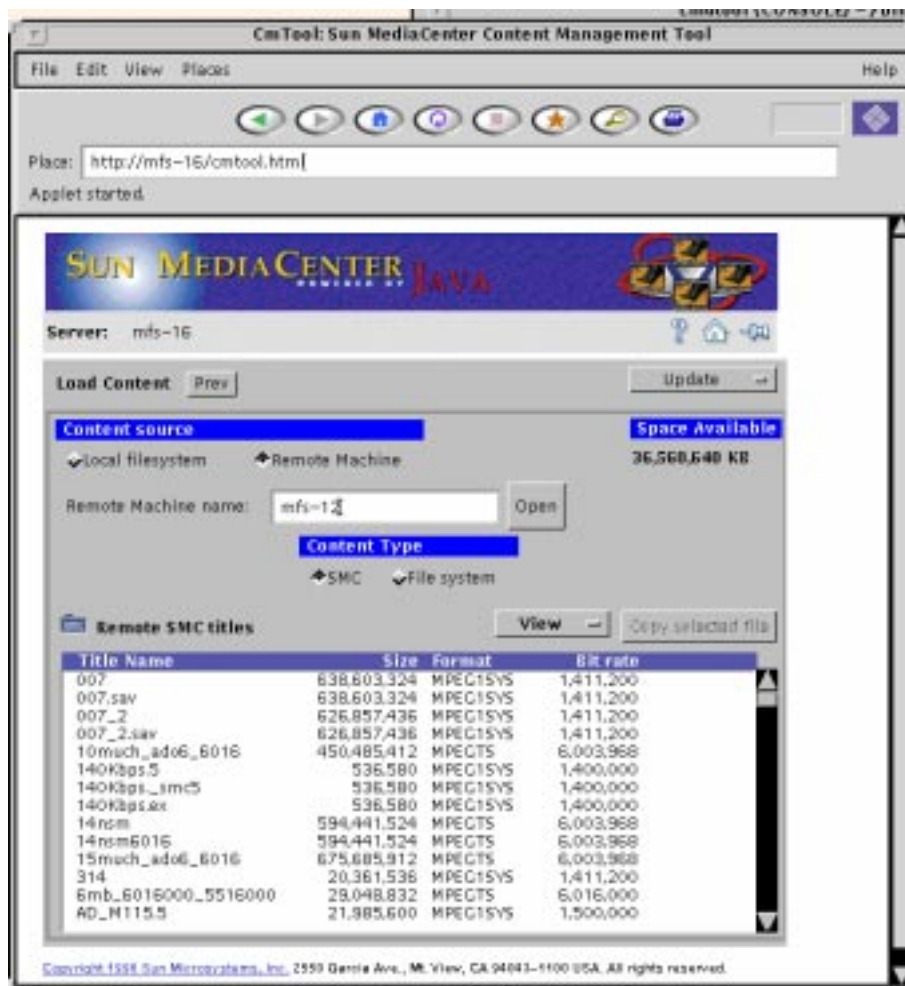


FIGURE 4-5 Content on Remote System

To copy a title from a remote system or local filesystem, select the title or file from the scrolling list, then click the Copy selected file button. (You can also display information about a selected title or file with the View button.)



You are prompted to specify a title for the content on the server, as shown in FIGURE 4-6. You can enter a name that is different from the source or use the same title. Remember that you cannot use spaces in titles. Click the Copy button (or select the Cancel button to cancel the copy operation).



FIGURE 4-6 Specifying a Local Title

You are prompted to confirm the copy operation, as shown in FIGURE 4-7. Click the `smc_copy` button (or select the Cancel button to cancel the copy operation).



FIGURE 4-7 Copy Confirmation

The Session Information page, shown in FIGURE 4-8, shows the status of the content transfer.

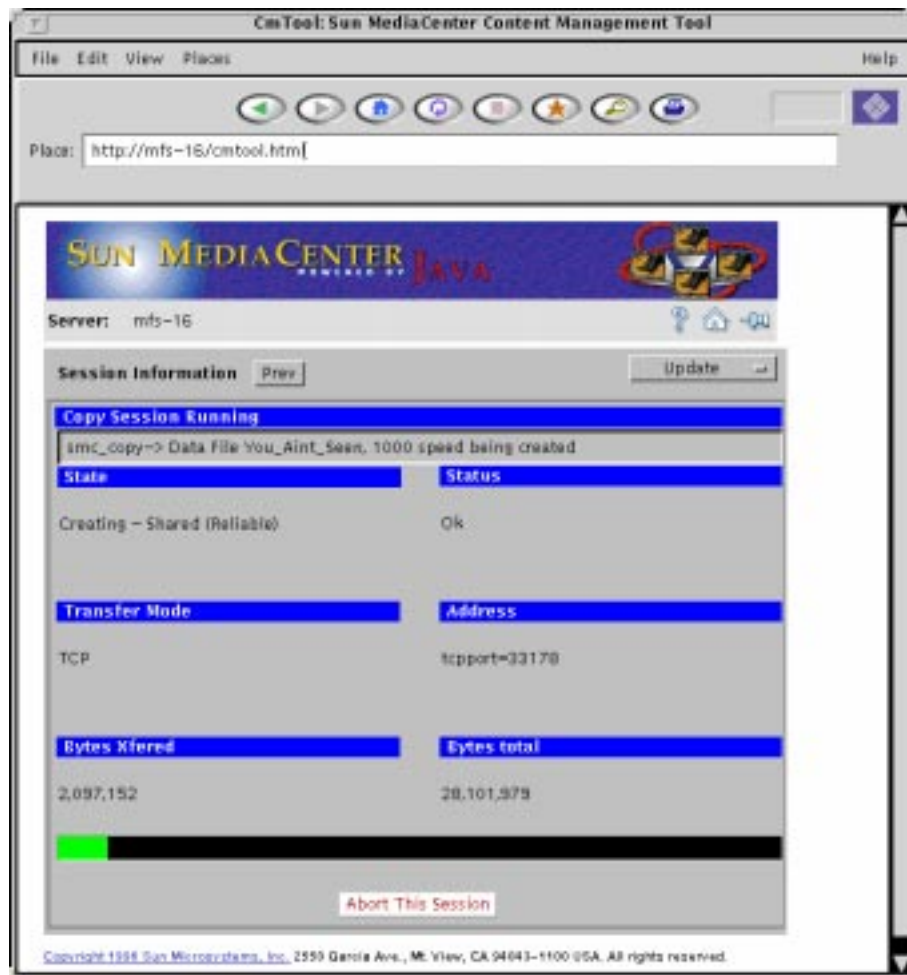


FIGURE 4-8 Session Information

A message appears when the content has been successfully copied to the server. The content is now available for viewing on the Sun MediaCenter server.

---

## 4.4 Table of Contents Page

This section describes the View and Selected menu options on the Table of Contents page.

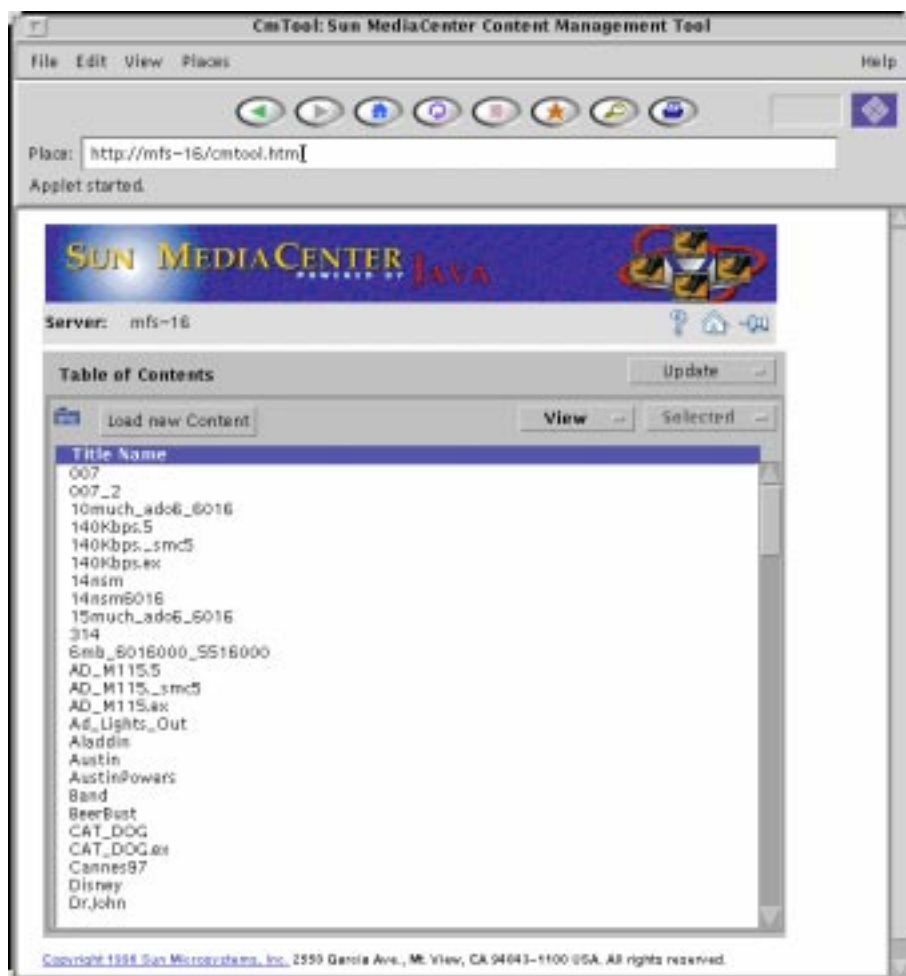


FIGURE 4-9 Table of Contents Page

## 4.4.1 View Menu

The View menu provides options that affect the display of titles in the Table of Contents page. The following options are available:

- Display allows you to select the attributes that you want displayed for the titles. For example, in addition to title name, you can specify that format and playtime be displayed. By default, only the title names are displayed in the Table of Contents page. When you select the Display option, the following dialog box appears:



FIGURE 4-10 Display Attributes Dialog Box

To display an attribute, click on the attribute name in the Hide Attributes list and then click on the Add >> button. To remove an attribute for the display, click on the attribute name in the Show Attributes list and then click on the << Remove button. Click the OK button to display the specified attributes in the Table of Contents page. Click Reset to retain the last-saved attributes or click Cancel to dismiss the dialog box with no change.

- Sort allows you to specify the sort order (ascending or descending) of the titles, based on a specific attribute. For example, you can display titles in order of ascending playtime (shortest playtime listed first). By default, the titles are displayed in alphanumerical order by name. When you select the Sort option, the following dialog box appears:



FIGURE 4-11 Sort Dialog Box

To sort by one or more attributes, click on the attribute in the Available Attributes list and then click on the Add >> button. To remove an attribute from the sort list, click on the attribute name in the Sort Order list and then click on the << Remove button. Select Ascending or Descending. Click the OK button to display the specified attributes in the Table of Contents page. Click Reset to retain the last-saved attributes or click Cancel to dismiss the dialog box with no change.

- Filter allows you to restrict the titles that are displayed. For example, you can display only those titles that have the format MPEG1SYS. By default, all titles are displayed. When you select the Filter option, the following dialog box appears:

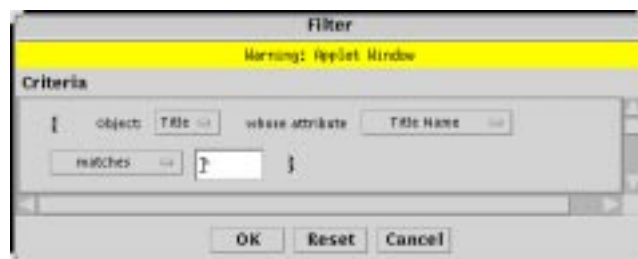


FIGURE 4-12 Filter Dialog Box

Select the title attribute on which the pattern match is to be done. Select either “matches” or “does not match,” then enter in the pattern that is to be matched/not matched. Click the OK button to filter the display in the Table of Contents page. Click Reset to retain the last-saved attributed or click Cancel to dismiss the dialog box with no change.

## 4.4.2 Selected Menu

The Selected menu allows you to view more information about a selected title, and to rename, delete, or play the title. Select a title in the Table of Contents page, then select the More Info option from the Selected menu. The Title Information page appears, as shown below:



FIGURE 4-13 Title Information Page

This page contains all available attribute information for the selected title. You can use the buttons at the bottom of the page to rename, delete, change the content description, or play the title.





## Sun MediaCenter Server Utilities

---

<i>Setting a Title Access Control List</i>	<i>page 5-1</i>
<i>The smc_copy Utility</i>	<i>page 5-4</i>
<i>The smc_tar Utility</i>	<i>page 5-8</i>
<i>The smc_ls Utility</i>	<i>page 5-17</i>
<i>The smc_rm Utility</i>	<i>page 5-21</i>

This chapter describes the utilities shipped with the Sun MediaCenter software.

Sun MediaCenter software offers a variety of means to transfer video files. See Section 2.4 “Content Loading” on page 2-5 for a discussion of the alternatives.

In this chapter and elsewhere in this document, a *title* refers to a piece of video content, such a movie. A title is stored on a Sun MediaCenter server as the set of files that make up a piece of content. The name of a title is but one attribute of a piece of content. When you specify a title name for file transfer or for removal, all attributes of that title are transferred or removed. See Section 6.2 “Video-File Attributes” on page 6-4 for a complete description of title attributes.

---

### 5.1 Setting a Title Access Control List

The Sun MediaCenter server software enables you to set an access control list (ACL) on a per-title basis. This type of ACL, along with server ACLs for titles and Content Manager sessions are described in Chapter 9 “Access Control Lists.”

When you create a new title on a Sun MediaCenter server by, for example, using `smc_copy` to copy a video file from one server to another, you (the copier) have read, write, and admin permissions for that title. Depending on the settings of the server ACL, other users can list your title (`smc_ls`) and get certain statistics on the title. However, a user besides you cannot copy your title, delete it, or append to it.

To extend access for a title that you own, use the `smc_settacl` command. `smc_settacl` is analogous to the Solaris `setfacl` utility. `smc_settacl` has the following syntax:

```
smc_settacl -s|m <acl_entries> [server:]<titlename> ...
smc_settacl -d <title_users> [server:]<titlename> ...
smc_settacl -f <filename> [server:]<titlename> ...
```

For the `-s` (set) and `-m` (modify) options, `<acl_entries>` stands for a comma-separated list of items of the form:

```
u[ser]:<username>:<permissions>
```

`<username>` is a Solaris login name; `<permissions>` is one or more of `r`, `w`, and `a` (read, write, and admin, respectively). In a `smc_settacl` command, you must specify three permissions. Replace any permission you are not setting with a hyphen. So, for example, if you are setting read and admin permission for the user Raj, any of the following ACL entries is valid:

```
user:raj:r-a
user:raj:-ra
user:raj:ar-
```

Permissions are defined in TABLE 5-1.

**TABLE 5-1** Title ACL Semantics

Permission	Title ACL
Read	Read this title data. Get statistics for this title.
Write	Write/Append this title data.
Admin (owner)	Delete/rename this title. Change ACL for this title.

For the `-d` (delete) option, `<title_users>` stands for a comma-separated list of items of the form:

```
u[ser]:<username>
```

...where `<username>` is a Solaris login name.

The `-s`, `-m`, and `-d` options accept an asterisk in the `<username>` field, so that you can specify ACL changes for all users on a server. You must use a backslash (`\`) with an asterisk, to prevent the shell from expanding the asterisk to mean the files in the local directory.

For the `-f` option, `<filename>` is a text file containing a list of entries of the form of `<acl_entries>`, above. Comments in the file, indicated by a hash mark (`#`) in column 1, are ignored by `smc_settacl`.

You cannot use the `-s` and `-f` options with any other option. You can combine `-m` and `-d`. The `-s` option replaces the current title ACL with the entry or entries you specify. The `-m` option modifies the title ACL. Using `-m`, if you specify a user who is not in the title ACL, that user is appended to the ACL. If you specify a user who *is* in the ACL, that user's permissions are changed to what you specify.

For all options, you can specify one or more title names, any of which can be local or remote. Separate multiple title names with a space between each pair. For a remote title, you prepend the name of the Sun MediaCenter server and a colon to the title name. You can use an asterisk in the title name field to stand for all titles on a server.

`smc_settacl` has a companion command, `smc_gettacl`, which displays the ACL for a specified title in the format used by `smc_settacl`. `smc_gettacl` is useful in assigning the values in one title's ACL to another title. For example, after you copy titles from the server `adam` to the server `eve`, you want to give your colleague Raj permissions to the files you just copied. Raj has `r, w, a` permissions on the title `ben_hur`. You enter:

```
adam% smc_gettacl ben_hur | smc_settacl -f - eve:\*
```

As with `smc_settacl`, `smc_gettacl` accepts multiple, space-separated title names.

Every title must have at least one user with admin permission. If you attempt to set or modify a title ACL such that all users with admin permission would be removed or have their admin permission removed, `smc_settacl` retains the last user with admin permission.

---

## 5.2 The `smc_copy` Utility

The `smc_copy` utility copies content, specified by a title name or a local filesystem name, between a local filesystem and a Sun MediaCenter server or between two Sun MediaCenter servers. Examples of a local filesystem are the Unix filesystem (UFS) and NFS™. `smc_copy` offers a convenient means to load content onto a Sun MediaCenter server and to backup content from the server to a local filesystem.

The `smc_copy` utility's specific functions are as follows:

- Copies a video file (such as an MPEG file) from a local filesystem to a Sun MediaCenter server. In so doing, performs activities required to make that title available for playing.
- Copies a video title from a Sun MediaCenter server to a local filesystem. Output is a tar file containing a content package. A title is a video file stored on a Sun MediaCenter server so that it can be played from the server. Names of titles specified in `smc_copy` commands are in the form returned by `smc_ls` (see Section 5.4 “The `smc_ls` Utility” on page 5-17).
- Copies a content package in a tar file to a Sun MediaCenter server. Input can be a pathname that includes a tar device. See Appendix B for a description of a content package.
- Copies a file between two local filesystems.

You can invoke `smc_copy` on one machine to copy content from a second machine to a third machine. The machine on which you invoke `smc_copy` does not have to be a Sun MediaCenter server. To run `smc_copy`, you need only the `smc_copy` binary, which you can copy from `/opt/SUNWsms/bin` on a Sun MediaCenter server with the current release of software.

The general form of `smc_copy` is:

```
smc_copy [-p] [-r rate] [-f format] [-s speed] [-t type] [-i] [-m] [-o] source [destination]
```

...where *source* and *destination* can be one of:

### *title*

The name of a title, such as `Heidi`.

Specifying only title implies that a title is stored on or being copied to the machine from which you are invoking `smc_copy`.

### *hostname:title*

The IP hostname of a Sun MediaCenter server and a title, for example, `guignon:heidi`.

*filename*

The name of a file in the form specified by a local filesystem, such as UFS or NFS. The filename might specify an MPEG file or, for a content package, a tar file.

*hostname:filename*

The IP hostname of a machine on which *filename* is stored, plus the *filename*.

You must supply an argument for *source*. Unless you use the `-m` option (for tar file input), you must also supply an argument for *destination*. If you are copying a title from a remote to a local machine, and not renaming the title, you can simply specify the name of the title as the destination.

`smc_copy` has the following options:

`-f {MPEGTS|MPEG1SYS}`

Indicates format of stream to be loaded onto Sun MediaCenter server (MPEG Transport Stream or MPEG-1 System Stream). There is no default for this option. By omitting this option, you implicitly request that the server software auto-detect the title's format. (Sun MediaCenter servers of version number 2.1 and later support format auto-detection.)

`-i`

Indicates input is from a Unix filesystem.

`-m`

Indicates input is a tar file. Used for loading a content package.

`-o`

Indicates output is to a Unix filesystem.

`-p`

Preserve the creation time in the destination file. Otherwise, the creation time becomes the current time for the newly copied file.

`-r rate`

Indicates the encoding bit rate for a title. There is no default for this option.

`-s speed`

For titles that contain trick play streams. *speed* is 1000 for a normal-play, forward-direction stream. The speed value for a stream of four times normal speed is 4000. Reverse direction is indicated by a minus sign, so that a stream of seven times normal speed, in the reverse direction has a speed value of -7000. The default speed value is 1000.

**-t *type***

Indicates the type of the file, either `data` or `index`. A data file is an MPEG file. An index file is a file containing splice points for switching back and forth among trick play streams.

When copying a title to or from a Sun MediaCenter server, you must have appropriate permissions in the ACL portion of the copied title. (See Section 5.1 “Setting a Title Access Control List” on page 5-1 for an explanation of title ACLs.) In the course of completing the copy operation, `smc_copy` creates a new title on the destination Sun MediaCenter server. You can rename the destination title in an `smc_copy` command.

To load content onto a Sun MediaCenter server, you must have permission, as specified in the `ServerAcl` file. This file is described in Chapter 9 “Access Control Lists.”

When you copy a title to or load content onto a Sun MediaCenter server, you, the copier, own the title access control list for that file. Other users can play the title, but cannot copy it to another server or remove it. To extend access to the newly-copied file, use `smc_settacl`, as described in Section 5.1, “Setting a Title Access Control List.”

The following are examples of `smc_copy` use.

The following example copies the title `heidi` from the Sun MediaCenter server `server2` to the local Sun MediaCenter server, `server1`, renaming the title to `heidi_orphan`, in the process:

```
server1% smc_copy server2:heidi server1:heidi_orphan
```

The following command accomplishes the same function as the preceding:

```
server1% smc_copy server2:heidi heidi_orphan
```

The following command loads content from a local Unix filesystem on the Solaris (UFS) machine `host1` to the remote Sun MediaCenter server `server1`, renaming the title in the process:

```
host1% smc_copy -i -r 1568700 heidi.mpg server1:Heidi
```

In the preceding command, note that the `-f` (format) option is omitted, meaning the format is auto-detected. Speed (`-s`) defaults to 1000 and type (`-t`) defaults to `data`.

The following command performs the same function as the preceding, except the `-t` and `-s` options are made explicit:

```
host1% smc_copy -i -t data -s 1000 -r 1568700 heidi.mpg server1:Heidi
```

The following command copies the title `heidi` from Sun MediaCenter server `server1` to a Solaris machine `host1`:

```
server1% smc_copy -o heidi /net/host1/export/mpeg_files/heidi.tar
```

The preceding command copies all files associated with the title `heidi`. If the title contains multiple streams for trick play, those streams, plus an index file for each stream, are copied to the tar file.

The following command is invoked from a third-party machine, `machine_x`, which is not an Sun MediaCenter server. The command loads content from an NFS file to the Sun MediaCenter server1.

```
machine_x% smc_copy -i -b 1587000 /net/host1/export/mpeg/oliver_twist.mpg \
server2:dickens_story
```

In the preceding command the speed and file type options are defaulted; the format is auto-detected by `server2`.

The following command copies content from a Sun MediaCenter server, `server1`, to a local filesystem, mounted on the third-party machine, `machine_x`:

```
machine_x% smc_copy -o server1:Heidi /net/host1/export/mpeg_files/heidi.tar
```

The following command copies a file between two local filesystems:

```
machine_x% smc_copy -i -o /home/mpeg/heidi.mpg /net/host1/export/mpeg/heidi.mpg
```

---

## 5.3 The `smc_tar` Utility

If you cannot use `ftp` to load content, as explained in Section 2.4 “Content Loading” on page 2-5, you can use `smc_tar`. The `smc_tar` command extracts a *content package* from a `tar` archival device or `tar` file. `smc_tar` parses the *TOC file* in the content package and stores the data and index files associated with the title, as well as the *TOC file* itself, into the correct locations on the Sun MediaCenter server.

`smc_tar` also works in the reverse direction, copying titles from a Sun MediaCenter server to a `tar` device or file.

`smc_tar` works with one or more local or remote Sun MediaCenter servers. For example, in a single command, you can copy titles from multiple local and remote Sun MediaCenter servers to a local tape drive, for backup purposes. Detailed examples are provided later.

You can run `smc_tar` on any SPARC/Solaris machine. To run `smc_tar` remotely, you need only the `smc_tar` binary, which you can copy from `/opt/SUNWsms/bin` on a Sun MediaCenter server with the current release of software.

### 5.3.1 Syntax and Options

The syntax for `smc_tar` is:

```
smc_tar t|c|x[v][b][w] f <tar device> [<blksize>] [<server>:<title>]
```

Options are analogues of `tar` options and are described as follows:

**c**

Specifies creation of a `tar` file or copying a title from the Sun MediaCenter server to a `tar` device. Requires a source file argument of `<server>:<title>`.

**x**

Specifies extraction of a `tar` archival or file. Used for copying content packages onto the Sun MediaCenter server.

**b** `<blknum>`

Specifies the blocking factor that was used to create the `tar` device contents. `<blknum>` must be a multiple of 20; one block equals 512 bytes. The recommended `<blknum>` value is 500 (256000 bytes). Most operating systems, including Solaris, have a default block number value of 20. The location of `<blknum>` in the `smc_tar` command line depends on the order of the `b` and `f` options. `<blknum>` is



the last argument in the command line if the `b` option is after the `f` option. `<blknum>` immediately follows the option list if the `b` option precedes the `f` option.

`w`

Prompts you before overwriting already-existing content.

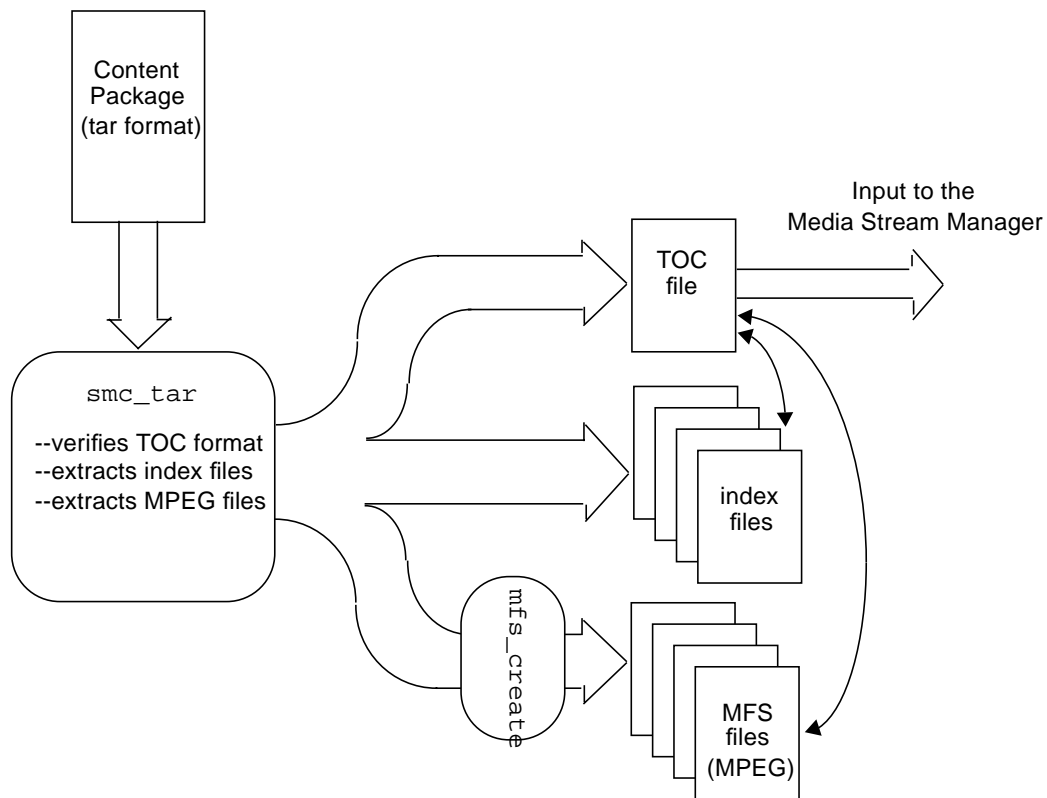
The `f` argument, which precedes a source `tar` file or device, is required for all `smc_tar` commands.

When writing to a `tar` file or device (`c` option), you are moving data from an MFS to a `tar` device. When extracting from a `tar` file or device (`x` option), you are moving data from a `tar` device to an MFS.

## 5.3.2 Content Loading

Before performing its file-loading function (`x` option), `smc_tar` performs parsing of the TOC file and error checking on the index files. If it detects a value in one of these files that implies an encoding error in a data file, it returns an error and exits. If `smc_tar` proceeds with and completes file installation, it indicates that the content has been formatted correctly for use on the Sun MediaCenter server.

The content-loading function of `smc_tar` is illustrated in FIGURE 5-1.



**FIGURE 5-1** Loading Content with `smc_tar`

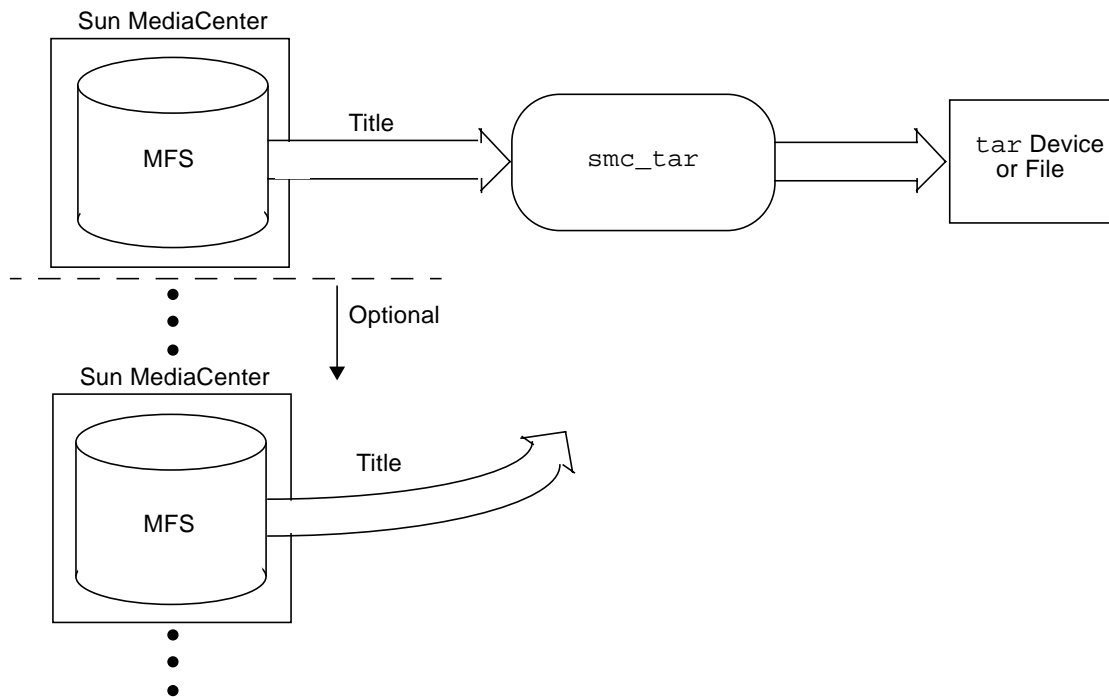
After loading content with `smc_tar`, you can pass a TOC file (title name) as an argument to MSM Client API functions (described in the *Sun MediaCenter Server Programmer's Guide*), to play the title described by that TOC file.

If a disk failure occurs when `smc_tar` is writing a title to a disk controlled by the Media File System (MFS), all files associated with that title—including the TOC file and index files—are erased.

### 5.3.3 Copying Content

With the `c` option, `smc_tar` copies all of the files associated with a title from one or more Sun MediaCenter servers to a tar device or file. This operation is illustrated in FIGURE 5-2.

Using `smc_tar`, you can copy files that were loaded onto the server by means other than `smc_tar`, such as with `smc_copy` or `ftp`.



**FIGURE 5-2** Copying a Title from a Sun MediaCenter Server

As you can infer from the figure, `smc_tar` provides a convenient means to backup titles on a Sun MediaCenter server.

---

**Note** – When backing up your video files, in addition to copying files from the disk subsystem used for the MFS, you should make a separate backup of the `/var/opt/SUNWsms/content` and `files` directories, which are stored on your server's (UFS) system disk. These directories contain files required by the MFS. Both `smc_tar` and `ftp` (described in Chapter 6) backup these required files in the course of backing up the MFS. However, if your system disk fails and the disk subsystem used for the MFS does not, you will find it more convenient to restore only the `content` and `files` directories than to restore the entire set of video files in the MFS.

---

## 5.3.4 Remote Copying and Extraction

You can run `smc_tar` from a Sun MediaCenter server or from a remote host that has a network connection to a Sun MediaCenter server.

The syntax for the `c` option is:

```
smc_tar cf <tar device> <server>:<title>
```

For the `c` option, you must have only one argument of the form `<server>:<title>`. `<server>` can be a local or remote Sun MediaCenter server and `<title>` can be the name of a specific title, with or without the `.TOC` extension.

The syntax for the `x` option is:

```
smc_tar xf <tar device> [<server>:<title>]
```

To extract the same title to multiple servers, you must enter an `smc_tar` command for each server:

```
remote_host# smc_tar xf /dev/rmt/0 smc_server1:bambi
remote_host# smc_tar xf /dev/rmt/0 smc_server2:bambi
```

## 5.3.5 Examples

The following are examples of `smc_tar` usage:

```
server# smc_tar xf /dev/rmt/0
```

The preceding command extracts a content package from the tape device `/dev/rmt/0`. Upon successful parsing of the TOC file and error checking of the index and data files, those files are stored on the server. This command causes files of the same name to be overwritten.

```
server# smc_tar xwb 40 /dev/rmt/0
```

The preceding command differs from the one before it in two respects, indicated by the `w` and `b` options. The `w` option means that you are prompted before files of the same name are overwritten. The `b` option, in combination with the value of 40, specifies that `smc_tar` uses a block size of 40 rather than the default of 20.

You can use `smc_tar` with the `x` option from a remote host, extracting content packages to a Sun MediaCenter server:

```
server# smc_tar xf /dev/rmt/0 smc_server:bambi
```

The following command shows how you can use the `c` option of `smc_tar` from a machine remote from a Sun MediaCenter server.

```
remote_host# smc_tar cf /dev/rmt/0 smc_server:bambi
```

The preceding command copies the title “Bambi” on the server `smc_server` to a local tape device. This command can be extended as follows:

```
remote_host# smc_tar cf /dev/rmt/0n smc_server1:bambi  
remote_host# smc_tar cf /dev/rmt/0n smc_server2:flipper
```

The preceding command copies the title “Bambi” on `smc_server1` and the title “Flipper” on `smc_server2` to a local tape device.

Because of the size of the files containing video content, it is usually preferable to create the `tar` file for a content package as piped input to `smc_tar`, rather than creating the `tar` file, then invoking `smc_tar` as a separate command. For example, on the Sun MediaCenter server, enter:

```
server# tar cf - <your title>.TOC <your content>.mpg | smc_tar xf -
```

Note that the TOC file is the first argument to `tar`. This is a requirement of a content package. The preceding command presumes that the TOC and content files are accessible from the Sun MediaCenter server. Where these files are remote from the Sun MediaCenter server, you can use a command such as:

```
remote_host# tar cf - <your title>.TOC <your content>.mpg | smc_tar xf \  
- smc_server:<your title>
```

A useful technique for situations in which your MPEG file resides on CD-ROM is to create a symbolic link from the directory where your TOC file resides to the file on CD-ROM. You then use `tar` with the `h` option, which causes `tar` to follow symbolic links. The following are example commands:

```
server# ln -s /cdrom/cdrom0/<your content>.mpg <your content>.mpg
server# tar cfh - <your title>.TOC <your content>.mpg | smc_tar xf -
```

An alternative to the preceding commands is:

```
server# tar cf - <your title>.TOC -C /cdrom/cdrom0 <your content>.mpg | smc_tar xf -
```

You can use commands such as those shown above to copy files to a remote Sun MediaCenter server. For example:

```
remote_host# ln -s /cdrom/cdrom0/<your content>.mpg <your content>.mpg
remote_host# tar cfh - <your title>.TOC <your content>.mpg | smc_tar xf \
- smc_server:<your title>
```

Examples of scripts that use `smc_tar` are shown below. The following script archives content to the locally-mounted `/dev/rmt/0` device and writes the content titles to the UFS file `SMCBackup.current-date` in the current directory.

```
#!/bin/sh
# smc_backup: Script to archive multiple SMC titles to tape
#
# To use, type
#   smc_backup [title]+
#
PATH="/sbin:/usr/sbin:/bin:/usr/bin:/usr/ccs/bin:/opt/SUNWsms/bin"
date=`date +%Y.%m.%d`
outputDevice="/dev/rmt/0"
contentsFile="SMCBackup.$date"
tapeNumber=1;
titleNumber=1;

mt status -f $outputDevice || exit 1

while [ $# -gt 0 ]; do
    title=$1;
    if smc_tar cf ${outputDevice}n $title; then
        echo "$date $tapeNumber $titleNumber $title" >> $contentsFile
        titleNumber=`expr $titleNumber + 1`;
        shift;
    else
        echo "Tape full; Rewinding ... \c"
        mt -f ${outputDevice} rew
        echo; echo "Insert new tape, hit return when ready \c";
        read answer
        tapeNumber=`expr $tapeNumber + 1`
        titleNumber=1
    fi
done
mt -f $outputDevice rew

echo "Content is in $contentsFile"
```

The following script locates a specified content title from a previously-created SMCBackup.*date* file. It then prompts you to load the appropriate tape to restore the archived title from the locally-mounted /dev/rmt/0 device.

```
#!/bin/ksh
# smc_restore: Script to restore an SMC backup

usage() {
    echo "usage: smc_restore [-c ContentsFile] [title]+" >&2
}

printNumber() {
    case $1 in
        *1)echo "$1st";;
        *2)echo "$1nd";;
        *3)echo "$1rd";;
        *)  echo "$1th";;
    esac
}

date=`date +%Y.%m.%d`
inputDevice="/dev/rmt/0"
contentsFile="SMCBackup.$date"

if set -- `getopt "c:" $*`; then
    while [ $# -gt 0 ]; do
        case "$1" in
            -c) contentsFile=$2;
                shift 2;;
            --) shift;
                break;;
            -*) usage;
                exit 1;;
            *) break;
               ;;
        esac
    done
fi

for t in $*; do
    found=false;
    grep $t $contentsFile | while read date tape fileNumber title; do
        echo "$t is the `printNumber $fileNumber` title on the `printNumber $tape`
        tape created on $date"
```



```

echo "Restore it? (yes or no) \c"
read response < /dev/tty
if [ $response = "yes" ]; then
    found=true
    echo "Load tape dated $date and hit return when done \c"
    read response < /dev/tty
    if mt -f ${inputDevice} rew && mt -f ${inputDevice}n fsf $((fileNumber -
1)); then
        if smc_tar xf $inputDevice $t; then
            echo "$t restored"
        fi
    fi
fi
done

if [ "$found" = "false" ]; then
    echo "The title you requested was not found on any backup tape"
fi
done

```

## 5.4 The smc\_ls Utility

The `smc_ls` utility lists the titles available for playback on a Sun MediaCenter server. You can play these titles through the facilities of the Media Stream Manager.

To run `smc_ls` remotely, you need only the `smc_ls` binary, which you can copy from `/opt/SUNWsms/bin` on a Sun MediaCenter server with the current release of software. `smc_ls` has the following syntax:

```
smc_ls [-s server] [-bBdDFhlnNrSUvwX] [title...]
```

...where *server* is the default server used for specifying one or more *title* arguments. If you do not specify the `-s server` option, `localhost` is the default.

You specify *title* arguments in the form *server:pattern*, where *pattern* is a restricted regular expression that follows the rules for shell filename matching. If *server* is not specified in *title*, the default server (as specified by the `-s` option or `localhost`) is used for that *pattern*. If *pattern* is not specified, all of the titles on the specified server are listed.

The other `smc_ls` options affect the display and sort of the title output. As a general rule, lowercase options specify the type of sort to be performed, while uppercase options specify the display of an attribute. By default, sort options turn on the corresponding display option. Options are invoked from left to right to provide the effect of a major/minor key sort. If no sort option is specified, titles are displayed using the collating rules of the native language. A few options are shortcuts for other display options; however this does not preclude these options from being combined with other sort and display options. The following describes these options:

`-b`

Sort the output in bitrate order, with the lowest bitrate first. For titles with multiple streams, the bitrate used for sorting is the maximum bitrate of all the streams that comprise the title. This option implies the `-B` option.

`-B`

Display the bitrate(s) for each of the streams that comprise a title. Streams that have corresponding index files are indicated with a trailing asterisk (\*). This option turns off the `-D` and `-S` display options.

`-d`

Sort the output in disk usage order, with the lowest disk usage first. The space used for sorting is the aggregate of all of the streams that comprise the title. This option implies the `-D` option.

`-D`

Display the disk usage. For titles with multiple streams, the aggregate is printed first. This is followed by a list of the space consumed by each of the trick play streams; each item in the list is separated by a plus sign (+). This option turns off the `-B` and `-S` display options.

`-F`

Display the format of this title.

`-h`

Display `smc_ls` syntax.

`-l`

Display a long listing. This option is an aggregation of the `T`, `U`, `N`, `F`, and `S` options. Subsequent options may override or extend the defaults.

`-n`

Sort the output in Normal Play Time (NPT) order, with the lowest NPT first. This option implies the `-N` option.

`-N`

Display the NPT.

- r  
Reverse the order of the sort. This option applies across all other sort options.
- S  
Display the play speeds for the title. For titles with multiple speeds, each available speed is listed, separated by a comma (.). Streams that have a corresponding index file are indicated by a trailing asterisk (\*). This option turns off the B and D display options.
- t  
Sort the output in the order of last modification time, with the most recently-modified first. This option implies the -T option.
- T  
Display the time the title was last modified.
- U  
Display the current usage of the title. A title may be currently in use by the Content Manager (cm) and/or the Media Stream Manager (msm), or it may be FREE, which means it is not in use by either cm or msm.
- v  
Verbose; print a banner at the top of each column of information.
- w  
Wide listing, do not truncate title names. This option is only useful with the -x option.
- X  
Displays output in Sun MediaCenter server 1.3 format. This option implies the U, N, F, and S options and inserts additional new lines between items. In addition, any operands are assumed to be simple server names and should not include a colon or patterns.

The following example displays a “long listing” of the content on the local Sun MediaCenter server, sorted in order of most-recently-modified time:

```
client% smc_ls -lt
AustinPowers      Jul 17 15:23 FREE   00:02:18 MPEG1SYS 1000
SMC_Trick         Jul 17 02:26 FREE   00:03:17 MPEG1SYS
1000,4000,8000,-4000,-8000
roc13            May 25 22:09 FREE   00:00:14 MPEGTS   1000*
SMCnews_1.24Mbs   May 25 21:54 FREE   00:01:20 MPEG1SYS 1000
SMCnews_0.15Mbs   May 23 16:32 FREE   00:01:21 MPEG1SYS 1000
SMCnews_copy      May 23 16:32 FREE   00:01:21 MPEG1SYS 1000
sun_on_cnn        May 23 11:05 FREE   00:15:00 MPEG1SYS 1000
```

Note that 1000 represents normal speed, forward direction. A value of 4000 represents four times normal speed. The value -4000 represents four times normal speed in the reverse direction.

An asterisk appended to a speed indicates that there is an index file present for that speed. An index file is a list of the splice points in a bit stream. (See Appendix B for a complete description.) Some decoders require an index file to reliably perform trick play; that is, to switch back and forth among streams or to jump forward or backward within a stream.

The string FREE indicates that the title is not being used by either the Content Manager or the Media Stream Manager. cm means that the title is being copied, either to or from the server. (Because of the server’s playthrough feature, you can play a title even as it is being loaded onto the server.) msm indicates that the title is being played.

The following example displays all lowercase titles on the server my-server:

```
smc_ls my-server:"[a-z]*"
dvd_ad
example
jumanj
mnymkr2
netboy90
```

The following example displays a long list of the current titles on the local server, with the output reverse-sorted by modification time (most-recently modified time last) and bitrate information displayed (instead of play speed):

```
smc_ls -lrtB
pc_test1.5      Apr 01 10:06 FREE    00:02:56 MPEG1SYS 1500000
PC_TEST_1.4    Apr 01 10:30 FREE    00:02:56 MPEG1SYS 1400000
Scott_and_Anne Jun 11 20:30 FREE    00:03:32 MPEG1SYS 2236000
rabbit         Jun 11 20:39 FREE    00:00:30 MPEG1SYS 1378000*
ISGs_Mariah    Jun 21 12:13 FREE    00:02:48 MPEG1SYS 1660400
Gisele_Kerozene Sep 16 19:13 FREE    00:04:14 MPEG1SYS 1504000
v2kill         Oct 04 17:34 FREE    00:04:20 MPEG1SYS 1394400
YOUAINT14      Nov 06 10:08 FREE    00:02:40 MPEG1SYS 1401728
smc5           Nov 22 15:00 FREE    00:06:31 MPEG1SYS 1500000
ST_HISTORY     Dec 13 13:52 FREE    00:09:02 MPEG1SYS 2213600
Max_Headroom   Jan 23 16:13 FREE    00:02:16 MPEG1SYS 1406000
mnymkr2        Feb 20 14:12 FREE    00:01:58 MPEG1SYS 1446848
dvd_ad         Mar 01 18:15 FREE    00:09:59 MPEG1SYS 2009344
Eastwood       Mar 13 17:42 FREE    00:06:31 MPEG1SYS 1500000
```

---

## 5.5 The `smc_rm` Utility

The `smc_rm` utility removes titles from an MFS on a Sun MediaCenter server. The command takes as an argument one or more title names. `smc_rm` removes a specified title, including MPEG files, supporting files, and attributes referred to by that title name.

The following example removes all content associated with the titles “Bambi” and “Quo Vadis”.

```
server% smc_rm SUNW_bambi SGI_quo_vadis
```



## Sun MediaCenter Server FTP Daemon

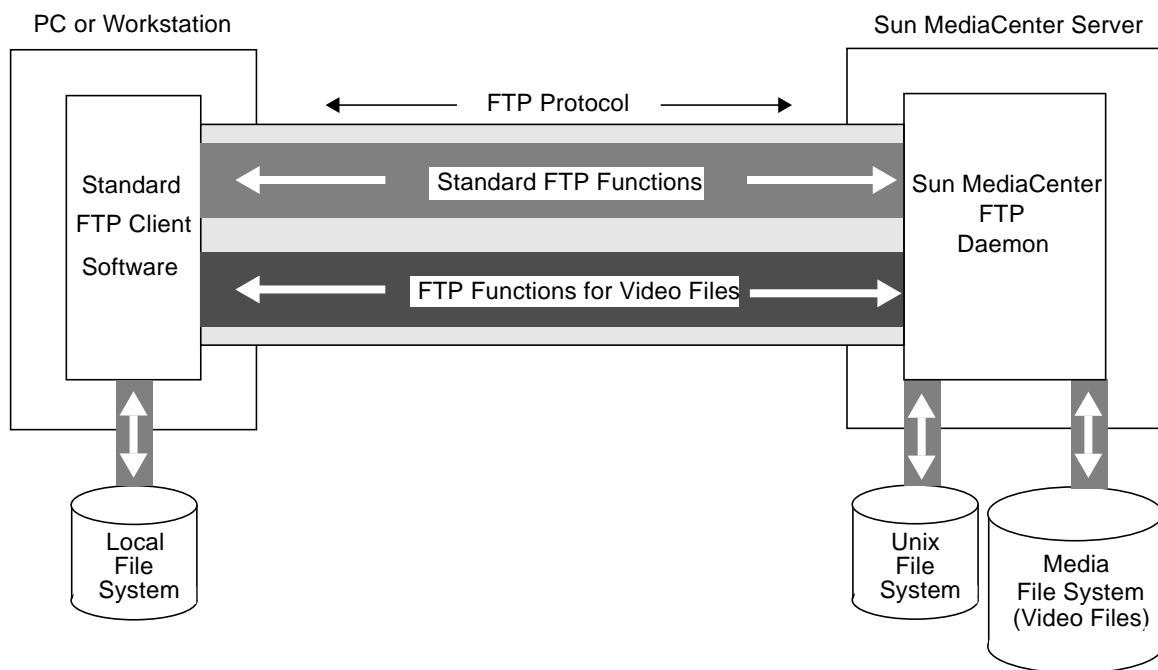
---

<i>Keyword for Video-File Functions</i>	<i>page 6-3</i>
<i>Video-File Attributes</i>	<i>page 6-4</i>
<i>Video-File Commands</i>	<i>page 6-5</i>
<i>Loading Content with ftp</i>	<i>page 6-14</i>

This chapter describes the Sun MediaCenter server FTP daemon. This daemon allows any `ftp` user—on any operating system/hardware platform—to access video files on a Sun MediaCenter server. In addition to this video server support, the daemon supports the “traditional” FTP functions described in the `ftp` (1) man page.

The video-file functions provided by the FTP daemon require no special client-side software. Also, the Sun MediaCenter FTP daemon supports any `ftp` client that conforms to the FTP protocol, regardless of the platform on which it is running.

FTP client’s support for video files is illustrated in FIGURE 6-1.



**FIGURE 6-1** Sun MediaCenter FTP Functions



---

## 6.1 Keyword for Video-File Functions

The keyword `smc:` in an `ftp` command invokes a video-file function. So, for example, to obtain a directory listing of the video files on a Sun MediaCenter server, you enter the following command:

```
ftp> dir smc:
Title           Usage      NPT          Format      Available Speeds
Bambi           cm,msm     00:03:42     MPEGPS      1000,
java_class      FREE       00:05:01     MPEGTS      1000,3000,-3000,
haircut         msm,       00:01:50     MPEG1SYS    1000,
list continues to conclusion...
ftp>
```

The output from `dir smc:`, as above, is identical to that of `smc_ls`, described in Chapter 5.

To obtain a listing of video files with all relevant attributes, enter:

```
ftp> ls smc:title=\*
```

You receive output such as the following:

```
smc:title=Bambi,format=MPEGPS,speed=1000,type=data,rate=3000000
smc:title=java_class,format=MPEG1SYS,speed=1000,type=data,rate=1500000
smc:title=haircut,format=MPEGTS,speed=1000,type=data,rate=2600000
list continues to conclusion...
ftp>
```

The attributes shown in the preceding output are described later.

Note that, at the `ftp` prompt, if you enter a `ls` or `dir` command without the `smc:` keyword, you receive a listing of the UFS files in the current working directory, just as you normally would. Standard FTP functions remain unchanged.

---

## 6.2 Video-File Attributes

A piece of video content is stored as a title on a Sun MediaCenter server. The `ftp` video-file commands report or perform operations on titles. A title is identified by the following attributes:

- name of the title, which usually corresponds to the name of the movie or video clip; for example, `Bambi` for the move “Bambi”. There is no default value for the `name` attribute.
- speed associated with the file. “Speed” refers to the speed and play direction of the title’s bit stream as compared to normal-play speed, forward direction. A value of `1000` indicates a file plays at normal play speed, in the forward direction. A value of `4000` indicates the file plays at four times normal speed, forward direction. A value of `-3000` indicates the file plays three times normal speed, in the reverse direction. The default value for the `speed` attribute is `1000`.
- type of the file, either data or index. A data file is usually a file containing an MPEG bit stream. An index file is a file that identifies the splice points within a bit stream. The Media Stream Manager (MSM) uses these splice points when stopping and restarting a stream and when performing trick play. See the *Sun MediaCenter Server Programmer’s Guide* for an explanation of trick play. The default value for the `type` attribute is `data`.
- rate at which the file containing the video bit stream was encoded. This is a number in the range from `1500000` to `8000000` that specifies a number of bits per second. The `rate` attribute applies only to data files, not index files. There is no default for the `rate` attribute.
- `format` is the format of the encoded bit stream. It can be one of:
  - `MPEGTS`, indicating an MPEG-2 Transport Stream.
  - `MPEGPS`, indicating an MPEG-2 Program Stream.
  - `MPEG1SYS`, indicating an MPEG-1 System Stream.
  - `MPEGTCE`, indicating an MPEG-2 encapsulation used by Thomson Consumer Electronics.

There is no default for the `format` attribute.

The fully-qualified name of a title consists of all of the just-described attributes. For example:

`title=Bambi,format=MPEGPS,speed=1000,type=data,rate=3000000`

The attribute values of the title named `Bambi` indicate a normal-play, forward-direction, MPEG file that was encoded 3 Mb/sec. in MPEG-2 Program Stream format.

All video-file commands require a value for the `title` attribute. (The value can be, with some commands, a wildcard.) Certain commands have requirements beyond this. These requirements are spelled out in the following section.

In some instances, such as when using video-file commands for backup, specification of all attributes is not required but is recommended. If you restore files that had not had their attributes recorded, you end up with files with default values for their attributes, which might not be appropriate for those files.

---

## 6.3 Video-File Commands

The `ftp` client commands you can use to access the FTP video-file functions on a Sun MediaCenter server are as follows:

- `delete`
- `dir`
- `get`
- `ls`
- `mget`
- `mls`
- `mput`
- `put`
- `rename`

The following subsections contain descriptions of each of these commands. The commands are categorized by function.

---

**Note** – The names used for `ftp` client commands are those used by the Solaris `ftp` client implementation. These command names might vary by implementation. For example, the `put ftp` command, described below, maps to the FTP protocol's `STOR` function. A given implementation might have a `copy` command that maps to the same function. For that implementation, the description of `put`, below, would apply to `copy`.

---

All `ftp` commands except `rename`, when used with the Sun MediaCenter FTP daemon, accept the asterisk (\*) as a wildcard character. When using the Solaris FTP client, you must use a backslash to escape the asterisk. Otherwise, the shell interprets the asterisk to mean all files in the current working directory. Other FTP clients, such as one you might use on a Wintel PC, might not have this requirement.

---

**Note** – The wildcard asterisk, as used with the Sun MediaCenter FTP daemon, has a different meaning from when you use the asterisk without the `smc:` keyword. As used with the Sun MediaCenter FTP daemon, the asterisk means “all of the titles on the server” or, with `mput`, “all of the titles in the current directory.” You can use the asterisk only by itself, not as a substitute for a piece of a string. For example, the following is valid syntax:

```
ftp> dir smc:title=\*
```

The example above, returns the names of all titles on the server. The following is *not* valid syntax:

```
ftp> dir smc:title=Bat\*
```

In the example above, the asterisk does not allow you to search on all titles that begin with the string “Bat”.

---

## 6.3.1 Listing Files

`dir`

The `dir` command lists the names of the data and index files associated with the titles on a Sun MediaCenter server. An example of its use:

```
ftp> dir smc:
Title      Usage      NPT      Format      Available Speeds
Bambi      FREE       02:03:42 MPEGPS      1000,
batman     cm,msm     01:38:06 MPEGTS      1000,
java_class cm,         00:45:01 MPEGTS      1000,3000,-3000,
haircut    msm,       00:12:50 MPEG1SYS     1000,
list continues to conclusion...

ftp> dir smc:title=batman
smc:title=batman,format=MPEGTS,speed=10000,type=index
smc:title=batman,format=MPEGTS,speed=10000,type=date,rate=3000000

ftp> dir smc:title=*
smc:title=Bambi,format=MPEGPS,speed=1000,type=data,rate=3000000,
smc:title=batman,format=MPEGTS,speed=10000,type=index
smc:title=batman,format=MPEGTS,speed=10000,type=date,rate=3000000
smc:title=java_class,format=MPEG1SYS,speed=1000,type=data,rate=1500000
smc:title=java_class,format=MPEG1SYS,speed=1000,type=index
smc:title=java_class,format=MPEG1SYS,speed=3000,type=data,rate=1500000
smc:title=java_class,format=MPEG1SYS,speed=3000,type=index
smc:title=java_class,format=MPEG1SYS,speed=-3000,type=data,rate=1500000
smc:title=java_class,format=MPEG1SYS,speed=3000,type=index
smc:title=haircut,format=MPEG1SYS,speed=1000,type=data,rate=2600000
list continues to conclusion...
ftp>
```

ls

Like the `dir` command, the `ls` command lists the names of the data and index files associated with the titles on a Sun MediaCenter server. An example of its use:

```
ftp> ls smc:title=*
smc:title=Bambi,format=MPEGPS,speed=1000,type=data,rate=3000000,
smc:title=batman,format=MPEGTS,speed=10000,type=index
smc:title=batman,format=MPEGTS,speed=10000,type=date,rate=3000000
smc:title=java_class,format=MPEG1SYS,speed=1000,type=data,rate=1500000
smc:title=java_class,format=MPEG1SYS,speed=1000,type=index
smc:title=java_class,format=MPEG1SYS,speed=3000,type=data,rate=1500000
smc:title=java_class,format=MPEG1SYS,speed=3000,type=index
smc:title=java_class,format=MPEG1SYS,speed=-3000,type=data,rate=1500000
smc:title=java_class,format=MPEG1SYS,speed=-3000,type=index
smc:title=haircut,format=MPEGTS,speed=1000,type=data,rate=2600000
list continues to conclusion...
ftp> ls smc:title=batman
smc:title=batman,format=MPEGTS,speed=10000,type=index
smc:title=batman,format=MPEGTS,speed=10000,type=date,rate=3000000
ftp>
```

mls

The `mls` command differs from the `ls` command in that it allows you to specify multiple arguments in a command line. `mls` assumes the last argument on an `mls` command line is the file that is to receive `mls` output. This argument can be a hyphen (-), indicating `stdout`. The syntax for `mls` is:

```
ftp> mls smc:title=file1 [smc:title=file2] ... <output file>
```

Examples of `mls` use:

```
ftp> mls smc:title=batman smc:title=seahunt -
200 PORT command successful.
150 ASCII data connection for smc:title=batman (129.144.90.32,56451).
smc:title=batman,format=MPEGTS,speed=10000,type=index
smc:title=batman,format=MPEGTS,speed=10000,type=date,rate=3000000
226 ASCII Transfer complete.
remote: smc:title=batman
242 bytes received in 0.045 seconds (5.2 Kbytes/s)
200 PORT command successful.
150 ASCII data connection for smc:title=seahunt (129.144.90.32,56585).
smc:title=seahunt,format=MPEGTS,speed=1000,type=data,rate=4000000
smc:title=seahunt,format=MPEGTS,speed=12000,type=data,rate=4000000
smc:title=seahunt,format=MPEGTS,speed=-12000,type=data,rate=4000000
226 ASCII Transfer complete.
remote: smc:title=seahunt
192 bytes received in 0.048 seconds (3.9 Kbytes/s)
ftp> mls smc:title=batman title=seahunt /tmp/mls_out
output to local-file: /tmp/mls_out? y (or carriage return)
200 PORT command successful.
150 ASCII data connection for /bin/ls (129.144.90.32,56735).
226 ASCII Transfer complete.
local: /tmp/mls_out remote: title=batman title=seahunt
238 bytes received in 0.021 seconds (1.8 Kbytes/s)
ftp>
```

## 6.3.2 Moving Files

When copying a data file (that is, an MPEG file), as distinguished from an index file, within `ftp`, use the `bin` command to put `ftp` in binary mode. Index files are ASCII files, so you can use ASCII mode, `ftp`'s default mode, for their transfer. When copying both data and index files, as you might with `mput` and `mget`, use `ftp`'s binary mode.

The multiple-copy commands, `mput` and `mget`, work as they do in their standard forms. If you invoke `ftp` with the `-i` option, you are prompted (`mget` or `mget <file>?`) for each file copied. If you invoke `ftp` without `-i`, `ftp` copies multiple files, one following the other, without intervening prompts.

For all commands described here, you must specify the name (`title=`) attribute. For `get` and `mget`, you must specify, in addition to name, speed and type. We recommend that you also specify rate and format, so that those values are available if need to copy the video file back to a Sun MediaCenter server.

With regard to video-file attributes, when you `get` or `mget` files, (the Media File System [MFS]-to-local file system direction) those files are stored in a local file system with as many or few attributes as you specify. Conversely, when you `put` or `mput` files (the local file system-to-MFS direction), those files are stored in the MFS with the attributes that were recorded in the local file system from which the files were copied.

FIGURE 6-2 illustrates the direction, between Sun MediaCenter server and a second machine, of the `put`/`mput` and `get`/`mget` commands.

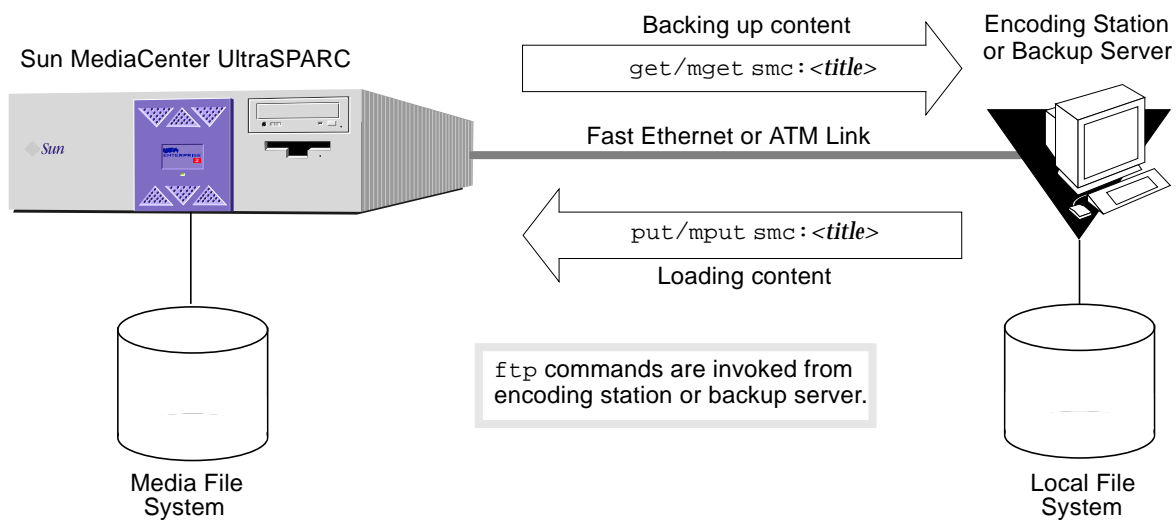


FIGURE 6-2 Directions of `put`/`get` Commands

### 6.3.2.1 Moving Single Titles

Use the `get` and `put` commands for individual titles. These commands do not accept the asterisk wildcard.



get

The `get` command allows you to copy a video file from a Sun MediaCenter server to a local file system. An example of its use:

```
ftp> bin
ftp> get smc:title=batman,format=MPEGTS,speed=1000,type=data,\
rate=3000000 /home/backup/batman.data
200 PORT command successful.
150 Binary data connection for smc:title=batman,speed=1000,type=data,
rate=3000000 (129.144.90.32,56133) (0 bytes).
226 Binary Transfer complete.
local: /home/backup/batman.index remote:
smc:title=batman,speed=1000,type=data,rate=3000000
1765800000 bytes received in 6237 seconds (2.7 Kbytes/s)
ftp>
```

put

The `put` command allows you to copy a video file from a local file system to a Sun MediaCenter server. It is the command you use when loading content onto the server. (See Section 6.4 “Loading Content with ftp” on page 6-14.) An example of its use:

```
ftp> put /home/backup/batman \
smc:title=batman,speed=1000,type=index,format=MPEG1SYS,rate=3000000
200 PORT command successful.
150 Binary data connection for smc:title=batman,speed=1000,type=data,
rate=3000000,format=MPEG1SYS (129.144.90.32,56133) (0 bytes).
226 Binary Transfer complete.
local: /home/backup/batman remote:
smc:title=batman,speed=1000,type=data,rate=3000000,format=MPEG1SYS
8900048 bytes received in 72 seconds (123.6 Kbytes/s)

ftp>
```

### 6.3.2.2 Moving Multiple Titles

The `mget` and `mput` commands are intended for the backup and restore of titles between the MFS and a local filesystem. You can use `mput` only with files that you have previously backed up with `mget`. For example, if you periodically use the command:

```
ftp> bin
ftp> lcd /home/backup/
ftp> mget smc:title=\*
```

...to back up your titles, you can use:

```
ftp> bin
ftp> lcd /home/backup/
ftp> mput \*
```

...to restore the backed-up titles. To allow the use of a subsequent `mput` command, the backup directory (`/home/backup`, in our example) must not contain any files other than titles backed up from a Sun MediaCenter server.

To summarize, `mget` can copy either a single title or all of the titles on a server; `mput` can copy only files that have been previously backed up with `mget`. The directory from which `mput` copies titles must contain only backed-up titles and no other files.

`mget`

The `mget` command allows you to retrieve a single title or all of the titles on a Sun MediaCenter server. `mget` supports the use of the asterisk (\*) wildcard character to specify all titles. An example of its use:

```
ftp> bin
ftp> lcd /home/backup/
ftp> mget smc:title=\*
200 PORT command successful.
<queries to confirm the transfer of each title>
ftp>
```

`mput`

The `mput` command copies files from a local file system to a Sun MediaCenter server. `mput` allows you to copy a single video title or all of the titles that have been copied to a given directory in a local file system using the `mget` command. `mput` supports the use of the asterisk (\*) wildcard character to specify all of the files in a directory. An example of its use:

```
ftp> bin
ftp> lcd /home/backup/
ftp> mput \*
200 PORT command successful.
<queries to confirm the transfer of each title>

ftp>
```

---

**Note** – You cannot use `mput` to load multiple titles onto a Sun MediaCenter server, unless those titles have been backed up using `mget`. To load multiple titles that have not been backed up (such as when loading titles for the first time), you must use multiple, individual `put` commands.

---

The directory containing the titles that you are copying with `mput` must contain only the titles being copied and no other files.

### 6.3.3 Renaming Files

`rename`

The `rename` command allows you to rename a video file on a Sun MediaCenter server. You specify only a title name for this command. All other attributes of the title remain the same.

An example of its use:

```
ftp> rename smc:title=batman
(to-name) smc:title=Batman

200 PORT command successful.

ftp>
```

Note that if a title contains trick play streams, these are renamed.

## 6.3.4 Deleting Files

`delete`

The `delete` (or `del`) command allows you to delete a video file from a Sun MediaCenter server. You specify only a title name for this command.

An example of its use:

```
ftp> del smc:title=Batman
250 DELE command successful.

200 PORT command successful.

ftp>
```

---

## 6.4 Loading Content with `ftp`

Use of `ftp` with the Sun MediaCenter FTP daemon is the preferred method to load content onto the server. You use the `ftp put` command to load content from an encoding machine or other machine on which content is stored to a Sun MediaCenter server. (See FIGURE 6-2 on page 6-10.) If a title contains trick play streams, you use a separate `put` command for the normal-play file and for each trick play file.

In loading content, you must specify the `title`, `speed`, `rate`, and `format` attributes for each MPEG file. The supported formats are `MPEG1SYS` and `MPEGTS`.

As an example, consider that you encode the title “Batman” at 3 Mbits/sec., in MPEG Transport Stream format. You have three MPEG files, one for normal speed, one for 21 times normal speed, forward direction, and one for 21 times normal speed, reverse direction. On the encoding machine, you use the following sequence of commands to load the title on to the server:

```
encoding_machine% ftp smc_server_one
Connected to smc_server_one.
220 smc_server_one FTP server (UNIX(r) System V Release 4.0) ready.
Name (smc_server_one:clinton): clinton
331 Password required for clinton.
Password:
230 User clinton logged in.
ftp> lcd /home/encode
ftp> bin
200 Type set to I.
ftp> put batman.1x.mpg smc:title=batman,format=MPEGTS,speed=1000,\
rate=3000000
ftp> put batman.21x.mpg smc:title=batman,format=MPEGTS,speed=21000,\
rate=3000000
ftp> put batman.n21x.mpg smc:title=batman,format=MPEGTS,speed=-21000,\
rate=3000000
```

Following completion of the put commands, you can access “Batman” on the Sun MediaCenter server by simply specifying `batman`. For commands such as `smc_copy`, or for the Media Stream Manager, the name `batman` identifies all of the files associated with that title, as well as all of the title’s attributes.

The Sun MediaCenter FTP daemon invokes Content Manager functions to automatically create index files for MPEGTS-format streams. Automatic index file generation does not occur for MPEG1SYS trick play streams. See Section B.5 “Index File Requirements” on page B-10 for a discussion of index files.



## Server Output Data Format

---

<i>Introduction</i>	<i>page 7-1</i>
<i>ATM Encapsulation</i>	<i>page 7-2</i>
<i>ATM Addressing</i>	<i>page 7-4</i>
<i>Fast Ethernet Encapsulation</i>	<i>page 7-5</i>
<i>Fast Ethernet Addressing</i>	<i>page 7-7</i>
<i>Specifying Encapsulation</i>	<i>page 7-8</i>
<i>Differences Among Different Types of Links</i>	<i>page 7-9</i>
<i>References</i>	<i>page 7-9</i>

This chapter describes the format of the data that is output from the Sun MediaCenter Server. It also tells you how to specify output data encapsulation for the Media Stream Manager Client API. This API is covered in the *Sun MediaCenter Server Programmer's Guide*.

---

### 7.1 Introduction

The Sun MediaCenter server transmits data to the client using the “push” model, that is, data is pushed to the clients without any flow control. The transmitted data is metered out by the server so that it conforms to the time-critical characteristics of its encoding. The control of the data—what to send, when to send it, and where to send it to—is carried out by the server’s Media Stream Manager (MSM), described in the *Sun MediaCenter Server Programmer's Guide*. The encoding of the data that are input to the server is described in Chapter 2. The movement of video data is

controlled by the Content Manager (CM). Utilities that use the CM are described in Chapter 5 and Chapter 6. The CM server and client APIs are described in the *Sun MediaCenter Server Programmer's Guide*.

FIGURE 7-1 illustrates the separation of data and control flows with respect to the Sun MediaCenter server.

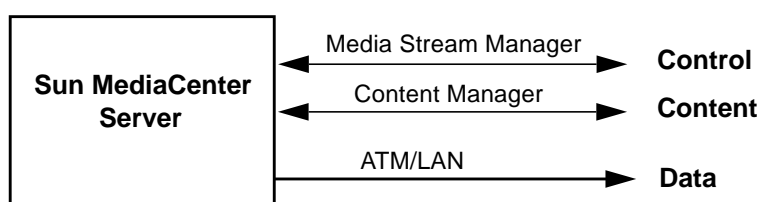


FIGURE 7-1 Separation of Data, Content Management, and Control

The server data output mechanism supports two types of network data links: SunATM and SunFastEthernet (100BaseT). The data formats and addressing for each type of link is described in the following sections.

---

## 7.2 ATM Encapsulation

This subsection describes the different ways the ATM data packets that are output from the server can be encapsulated.

### 7.2.1 MPEG-2 Transport Stream Packet Encapsulation

The encapsulation for MPEG-2 Transport Streams is the default specification of the standard public document from ATM Forum document 95-0012R5 (see reference [1] on page 7-9), titled *Service Aspects and Applications*. In this document, the chapter “Network Adaptation” describes the required default ATM packet for video to be made up of two MPEG Transport packets encapsulated in a single ATM AAL-5 packet. Each MPEG-2 Transport packet is 188 bytes long.



## 7.2.2 MPEG-2 Program Stream Packet Encapsulation

The MPEG-2 Program Stream can have any packet size. The server delivers AAL-5 packets with an MPEG data size corresponding to twenty MPEG Transport packets or 3760 bytes for the Program Stream encapsulation. This encapsulation uses a larger ATM packet size than is used for MPEG-2 Transport Streams.

---

**Note** – Sun is not committed to the packet encapsulation for the MPEG-2 program stream. We may choose a different encapsulation to, for example, accommodate larger packet sizes, in future releases.

---

## 7.2.3 MPEG-1 System Stream Packet Encapsulation

The MPEG-1 System Stream can have any packet size. The server delivers AAL-5 packets with an MPEG data size corresponding to twenty MPEG-2 transport packets or 3760 bytes for the Program Stream encapsulation. This encapsulation uses a larger ATM packet size than is used for MPEG-2 Transport Streams.

## 7.2.4 Thomson Electronics Packet Encapsulation

The encapsulation for Thomson Consumer Electronics (TCE) packets is the one required by TCE. This encapsulation consists of multiple 130-byte MPEG packets. These MPEG packets have a special TCE-specific header.

## 7.2.5 MPEG-2 Encapsulation over Classical IP over AAL5

The server encapsulates MPEG-1 and MPEG-2 packets as specified in RFC 1577, “Classical IP and ARP over ATM” and RFC 1483, “Multiprotocol Encapsulation over ATM.”

The server creates a UDP packet of a size that accommodates the maximum number of 188-byte MPEG packets that fit into a Media Transmission Unit (MTU). Consult the SunATM documentation for the MTU size for that product.

---

## 7.3 ATM Addressing

You specify the destination address with the MSMC API's `msmSetConnect()` function. You fill in an instance of the `MsmConnect` structure before calling `msmSetConnect()`. This structure contains an `destTiAddr` field, which holds a string. This field determines the destination address to which the server delivers data.

### 7.3.1 MPEG over AAL5

When configured for MPEG over AAL5, the server delivers video data to a specified ATM virtual circuit over a specified server ATM port. Therefore, the data endpoint address is described by a ATM virtual circuit and a server ATM port.

---

**Note** – The server supports only SunATM devices, that is, devices with names of the form `ba<num>`.

---

The address is a string of the following format:

<code>port=ba&lt;ATM portnum&gt;,vc=&lt;ATM vnum&gt;</code>
---

The components of this address are as follows:

- The `ba<ATM portnum>` is the server's ATM interface, through which multimedia data will pass.
- The `<ATM vnum>` specifies the ATM circuit that has been established to deliver the multimedia data through the ATM network to the client.

For example, the address `port=sa0,vc=301` specifies that data be delivered via the server's ATM device `sa0` to virtual circuit 301.

### 7.3.2 MPEG-2 over Classical IP over AAL5

When configured for MPEG over Classical IP over AAL5, the server delivers video data to a specified IP hostname and UDP port number.

The address is a string of the following format:

`host=<IP hostname>,udpport=<UDPport num>`

The components of this address are as follows:

- The `<IP hostname>` is the destination client's IP hostname.
- The `<UDPport num>` is the destination UDP port number on which the client host is awaiting data.

For example, the address of `host=cordelia,udpport=40000` specifies that data be delivered to host `cordelia` at UDP port `40000`.

To successfully reach a destination, the server must use ATM ARP or a local configuration file to resolve an IP address to an ATM address.

---

## 7.4 Fast Ethernet Encapsulation

This section describes the different ways the Fast Ethernet data packets that are output from the server can be encapsulated. The Fast Ethernet data distribution is intended for LAN environments and is illustrated in FIGURE 7-2.

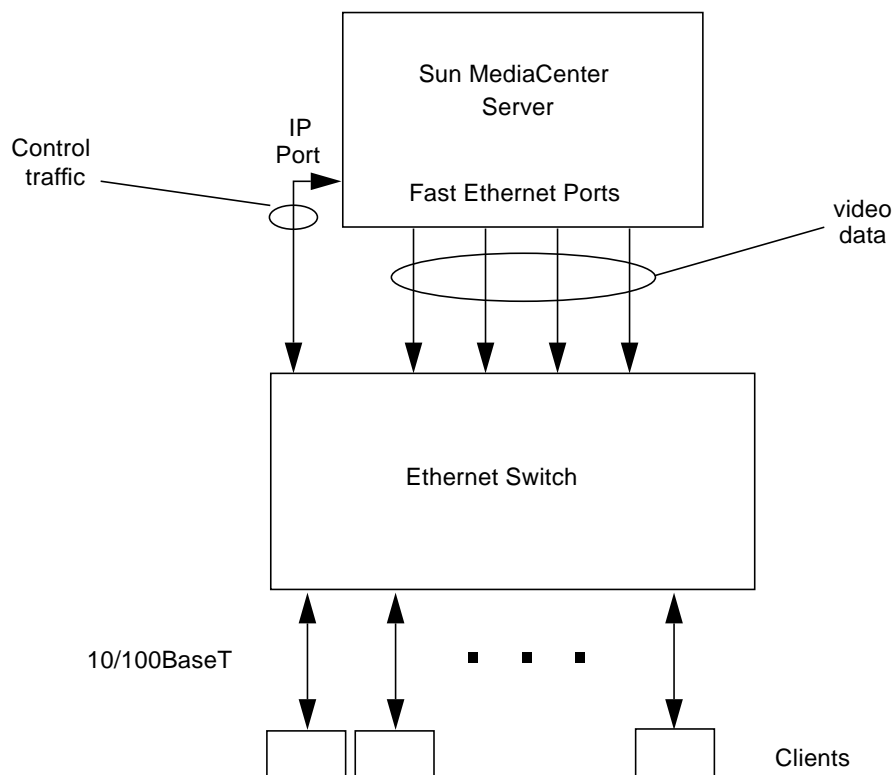


FIGURE 7-2 LAN/Enterprise Scenario

## 7.4.1 MPEG-2 Transport Stream Packet Encapsulation

As of the date of this publication, there is no standard for transmitting MPEG over Ethernet. The encapsulation supported by the Sun MediaCenter server consists of multiple MPEG-2 transport packets as payload of UDP over IP, which provides a connectionless transport mechanism (see RFC 768 for description of the UDP protocol). The number of MPEG transport packets for every UDP payload is seven, which is the maximum number of 188-byte MPEG-2 transport packets that fits into the maximum-size Ethernet packet of 1500 bytes.

## 7.4.2 MPEG-2 Program Stream Packet Encapsulation

The MPEG-2 Program Stream is not defined to have any fixed packet size by the MPEG-2 standard. The server delivers UDP packets with payload containing MPEG data of size corresponding to seven MPEG-2 transport packets or 1316 bytes for the program stream encapsulation. This encapsulation uses the same Ethernet packet size as is used for the MPEG-2 Transport Stream.

## 7.4.3 MPEG-1 System Stream Packet Encapsulation

The MPEG-1 System Stream is not defined to have any fixed packet size by the MPEG-1 standard. The server delivers UDP packets with payload containing MPEG data of size corresponding to seven MPEG-2 transport packets or 1316 bytes for the program stream encapsulation. This encapsulation uses the same Ethernet packet size as is used for the MPEG-2 Transport Stream.

## 7.4.4 Thomson Electronics Encapsulation

The encapsulation supported here consists of multiple TCE MPEG packets as payload of UDP over IP, which provides a connectionless transport mechanism. The number of TCE MPEG packets for every UDP payload is eleven, which is the maximum number of 130-byte TCE transport packets that fits in the largest allowable Ethernet packet of 1500 bytes.

---

## 7.5 Fast Ethernet Addressing

The server delivers video data to a specified IP hostname and UDP port number. Therefore, the data endpoint address is described by the concatenation of an IP hostname and UDP port.

You specify the destination address with the MSMC API's `msmSetConnect()` function. You fill in an instance of the `MsmConnect` structure before calling `msmSetConnect()`. This structure contains an `destTiAddr` field, which holds a string. This field determines the destination address to which the server delivers data. The address is a string of the following format:

`host=<IP hostname>,udpport=<UDPport num>`

The components of this address are as follows:

- The *<IP hostname>* is the destination client's IP hostname.
- The *<UDPport num>* is the destination UDP port number on which the client host is awaiting data.

For example, the address of `host=cordelia,udpport=40000` specifies that data be delivered to host `cordelia` at UDP port `40000`.

The server must have available a name service (or appropriate entries in local `/etc/hosts` and `/etc/ethers`) so that the client hostname can be resolved to the four-byte IP address and the six-byte Ethernet address before data transmission begins.

---

## 7.6 Specifying Encapsulation

For all types of output links, you specify the encapsulation of the data for a particular bit stream with the MSMC API's `msmSetConnect()` function. You fill in an instance of the `MsmConnect` structure before calling `msmSetConnect()`. This structure contains an `encap` field, which holds a string. This field determines the encapsulation that the server uses to deliver data to the client. TABLE 7-1 lists the strings used to specify different encapsulations.

**TABLE 7-1** Value of `encap` Field in `MsmConnect`

Encapsulation Type	<code>encap</code> Contents
MPEG-2 Transport Stream	MPEGTS
MPEG-2 Program Stream	MPEGPS
MPEG-1 System Stream	MPEG1SYS
Thomson Consumer Electronics	MPEGTCE

---

## 7.7 Differences Among Different Types of Links

The following are distinctions among the ways a Sun MediaCenter server handles video output over Fast Ethernet or ATM links:

- As described in this chapter, MPEG streams are encapsulated in UDP/IP packets and then in Ethernet packets, when sent over Fast Ethernet links. Over ATM, MPEG streams can be encapsulated in AAL5 ATM packets or in IP packets for subsequent encapsulation in AAL5 packets.
- A Sun MediaCenter server can treat multiple Fast Ethernet links as a single virtual pipe or as individual interfaces. In the virtual pipe (called “load sharing”) case, the server performs load balancing over multiple Fast Ethernet links. The first requested bit stream is directed over the first link. The next bit stream is sent over the next link, and so on, in round-robin fashion. By contrast, multiple ATM links are always addressed individually. A result of this is that an MSM client could, potentially, reach the bandwidth limit on a given ATM link before the server’s bandwidth limit was reached. In such a situation, the server stops accepting new requests to ensure timely servicing of existing requests. This situation of a single filled-up link causing a pausing of admissions could not occur on a Fast Ethernet server on which load sharing was in effect.
- How an MSM client specifies a destination differs among the different output types of the Sun MediaCenter server. For a Fast Ethernet model and for ATM servers that use Classical IP over ATM, the MSM client specifies a destination without regard to a specific interface. For ATM server that uses MPEG over AAL 5, the MSM client specifies a given ATM interface on the server, as well as a virtual circuit number that identifies the destination.
- By default, links used for video output are not available to TCP/IP traffic. As used by a Sun MediaCenter server, network interfaces used for video output are one-way, from server to destination. However, you can configure an ATM link to share bandwidth between two-way, TCP/IP traffic and video output traffic. Follow the instructions for Classical IP configuration over AAL5/ATM in the *SunATM SBus Cards Manual*.

---

## 7.8 References

[1] SAA Audio-visual Multimedia Service (AMS) Implementation Agreement, ATM Forum/95-0012R5, August 7-11, 1995.





## Administering MFS Disks

---

<i>Monitoring MFS Disks</i>	<i>page 8-1</i>
<i>MFS Utilities</i>	<i>page 8-2</i>
<i>Replacing a Failed Disk</i>	<i>page 8-10</i>

This chapter tells how to monitor MFS disks and how to detect and respond to disk failures.

---

### 8.1 Monitoring MFS Disks

The easiest way to monitor the Media File System (MFS) disks is to use the Sun MediaCenter WebAdmin tool. The WebAdmin tool is a Web-based graphical user interface (GUI) for monitoring the Sun MediaCenter server and is part of the software that is installed on the Sun MediaCenter server. Using the WebAdmin tool is discussed in detail in Chapter 3 “Monitoring the Server.”

The Disk Health page in the WebAdmin tool displays the disks in groups, as specified by the RAID layout. Individual disks can be either Data or Parity type. The name of each disk is shown, along with an icon that shows the status of the disk. Icons can be one of the following colors:

- green is normal
- red is failed or reconstructing

You can also monitor console output with the WebAdmin tool, including checking `/var/adm/messages` for console messages that might have scrolled off your screen or window.

You can also use the Solstice Domain Manager (or other SNMP-conformant program) to monitor the Media File System (MFS) disks. The SNMP agent for the Sun MediaCenter server, `sms-vod`, has an SNMP trap, `Disk_Failure`, that allows for automatic notification when a disk failure occurs. This trap is particularly significant if a disk failure is followed by a reboot of the server. If this occurs, your SNMP manager retains a record of the disk failure, while on the server, notification does not persist across a reboot (see note on page 8-12). You can also create an event request that will report when user-specified thresholds are met or exceeded for certain attributes. See Appendix A for discussion of the attributes of the video server MIB, as well as instructions for enabling trap handling in Domain Manager.

---

## 8.2 MFS Utilities

This section describes MFS utilities that you can use to display information about the MFS, create an MFS, or write data to a new data disk in the MFS. TABLE 8-1 lists these utilities.

**TABLE 8-1** MFS Utilities

Command	Function
<code>mfs df</code>	Displays disk information for the currently-mounted MFS.
<code>mfs diskusg</code>	Displays disk usage for a specified MPEG file.
<code>mfs repair</code>	Writes data onto a new data disk after failure of a data disk.
<code>mkmfs</code>	Creates the MFS. You need to invoke this utility only if you reconfigure the data and parity disks that make up an MFS.

If you install Sun MediaCenter software in the default location (`/opt/SUNWsms`), the path to the MFS command-line utilities is `/opt/SUNWsms/bin`. The exception to this is `mkmfs`, which requires a path of `/opt/SUNWsms/mfs/bin`.

### 8.2.1 `mfs df`

The MFS's `mfs df` utility is analogous to the Unix file system `df` (1B) utility. `mfs df` displays the amount of disk space occupied by currently-mounted MFS file system, the amount of used and available space, and how much of the file system's total capacity has been used. By default, the MFS's `df` utility reports space in increments of 64K blocks. With the `-k` option, `mfs df` reports space in kilobytes.

`mfs df` usage is as follows:

```
mfs df [-kv]
```

The `mfs df` command has the following options:

`-k`

Space is reported in kilobytes, rather than 64K blocks

`-v`

An option that produces “verbose” output.

The two options can be used together, to obtain verbose output with space reported in kilobytes.

The following is an example of `mfs df` usage, without the `-v` option:

```
server% mfs df
Total      Used      Avail  Capacity  Type
777216     231008     546208    30%    (data)
194304     54245     140059    28%    (parity)
```

In the output above, note that `mfs df` reports space information for both the data and parity disks.

```
server% mfs df -v
Blksize 64k, version 2
Device      Total      Used      Avail      Capacity  Type
/dev/rdisk/clt1d0s2  65408      3159      62249      5%    (data)
/dev/rdisk/clt3d0s2  65408      3159      62249      5%    (data)
/dev/rdisk/clt4d0s2  65408      3159      62249      5%    (data)
/dev/rdisk/clt5d0s2  65408      3159      62249      5%    (data)
/dev/rdisk/c2t1d0s2  65408      3157      62251      5%    (data)
/dev/rdisk/c2t2d0s2  65408      3157      62251      5%    (data)
/dev/rdisk/c2t3d0s2  65408      3157      62251      5%    (data)
/dev/rdisk/c2t4d0s2  65408      3157      62251      5%    (data)
/dev/rdisk/c2t5d0s2  65408      3157      62251      5%    (data)
/dev/rdisk/clt6d0s2  65408      3159      62249      5%    (parity)
/dev/rdisk/c2t6d0s2  65408      3157      62251      5%    (parity)
```

Note in the verbose `mfs df` output, above, that the MFS block size (64k) is reported, as well as the disk type, which can be either `data` or `parity`. Output with the `-k` option is the same except that the `Total`, `Used`, and `Avail` data is displayed in kilobytes rather than 64K blocks. The fact that space consumed is the about the same on all disks is characteristic of the MFS, in which data is striped across all disks.

## 8.2.2 `mfs diskusg`

The `mfs diskusg` utility returns information on disk usage of an MPEG file, given the size or duration, encoding rate, and encapsulation format that you supply. It has the following syntax:

```
mfs diskusg [-v] [-g]{size=size | duration=seconds} rate=rate format=format
```

The parameters are described as follows:

`-v`

Specifies verbose mode, in which `mfs diskusg` displays information that would be returned by the `mfs dump` utility, if a file of the specified parameters were in the MFS.

`-g`

Returns an error status if there is no space for the specified content. This parameter is intended for use in shell scripts where you need to determine if there is enough space to store a particular content. If you specify both the `-v` and `-g` parameters at the same time, only the `-g` option will take effect.

*size*

Size of the MPEG file, in bytes.

*duration*

Duration of the MPEG file, in seconds.

*rate*

Encoding rate of the MPEG file, in bits per second.

format=*format*

Format of the encoded source file. One of:

MPEGTS

MPEG Transport Stream packet format. The packetization of this encapsulation consists of two 188-byte packets.

MPEGPS

MPEG Program Stream packet format.

MPEGLSYS

MPEG-1 System Stream packet format.

MPEGTCE

MPEG Thomson Consumer Electronics format. The packetization of this encapsulation consists of an arbitrary number 130-byte packets.

The `mfs diskusg` utility is useful in conjunction with `mfs df`. You can compare the output of `mfs diskusg`, the amount of space a file would consume, with the output of `mfs df`, which tells you the space available.

The following is an example of the use of `diskusg`:

```
server% mfs diskusg size=1280000000 rate=4000000 format=MPEGTS
20480
```

In the example above, you obtain the number of 64K blocks consumed by a 1.28 GB file, encoded at 4 Mb/sec., and encapsulated in MPEGTS format. This is the number of data blocks, not parity blocks. Using the `-v` (verbose) option with the same supplied parameters, you receive:

```
server% mfs diskusg -v size=1280000000 rate=4000000 format=MPEGTS
Rate 4000000 erate 4000640 bsize 65536 lbsize 262144 ebsize
250040
Redundancy set size 4 zones 2 size 1280000000
Psize 188 Data drives 24 Parity Drives 6
Data blks 5120 Parity blks 1280
Data disk usage 20480 blks (1342177280)
Parity disk usage 5120 blks (335544320)
```

### 8.2.3 mfs repair

The `mfs repair` utility reconstructs data for a failed disk using available parity information, writing this reconstructed data to a new disk. It can also perform consistency checks on the MFS, including reclaiming unused blocks, and can be used to explicitly fail a disk.

You can use `mfs repair` while the Sun MediaCenter server is running and delivering video streams, or when the server is halted.

`mfs repair` usage is as follows:

```
mfs repair [-cs] [-Fpr] [-d /dev/rdisk/device] [-ovy]
```

The arguments to `mfs repair` are described below:

-c

Perform consistency checking, including reclaiming unused blocks.

-s

Display the status of a failed drive (if any).

-F

Force a failure of a specified disk.

-p

Initiate phase-out of parity mode. (Return to normal operation from parity mode.)

-r

Reconstruct data on newly replaced disk drive.

-d *device*

The SCSI controller and target numbers of the device to be rebuilt from parity information. An example device name is `/dev/rdisk/c1t3d0s2`.

-v

Verbose mode. List names of files as they are encountered.

-y

Answer “yes” to all questions.

Assume an example Sun MediaCenter server for which you receive an SNMP `Disk_Failed` trap. On checking console messages you see the following message:

```
WARNING: data disk 0x80009a (32,154) FAILED
```

---

**Note** – It is only after you receive a message such as the preceding that you can say that a disk has failed. An accumulation of “soft” errors, such as SCSI read errors, might be the harbinger of a disk failure, but does not of itself constitute a disk failure.

---

To find out which MFS disk has failed, you enter:

```
server# mfs repair -s
drive /dev/rdisk/clt3d0s2 (dev 0x80009a) is FAILED
```

`mfs repair` reports that the disk at `clt3d0s2` has failed. Chapter 3 of the *Sun MediaCenter UltraSPARC Hardware Manual* tells you that `clt3d0s2` corresponds to the third data disk in the first Multipack enclosure connected to SCSI port 1 on the server.

With the server still running, you remove the defective disk and replace it with a good disk. You then enter the following command to copy data from the parity disk to the new disk:

```
server# mfs repair -r -d /dev/rdisk/clt3d0s2 -v
```

The `mfs repair` utility takes several minutes to restore a new disk. When `mfs repair` finishes copying data to the new disk, the server is still in parity mode, that is, reading data from the parity disk and not using the new disk.

To make the server return to normal playback mode (phase out parity mode), enter the following command:

```
server# mfs repair -p -d /dev/rdisk/clt3d0s2 -v
```

`mfs repair` displays “Recovering” messages on the console as the server returns to normal playback mode.

You can combine the `-r` and `-p` options, shown in the previous two example commands. For example:

```
server# mfs repair -rp -d /dev/rdisk/clt3d0s2 -v
```

With a command such as the preceding, you can combine repair and return to normal playback in a single command.

The status of a disk as failed does not persist across a reboot. If you experience a disk failure, then reboot, you must explicitly fail the disk, with a command such as:

```
server# mfs repair -F -d /dev/rdisk/clt3d0s2 -v
```

Following this command, use the `mfs repair` commands described above to replace the failed disk.

Use of `mfs repair` is appropriate only for single disk failures. Two, overlapping disk failures cause the server to panic. If you experience the failure of more than one disk, you must use `smc_copy` or `ftp`, in conjunction with the Sun MediaCenter FTP daemon, to restore your original data from backup storage.

For a given disk, you might receive a number of “soft errors,” such as SCSI read errors that were corrected upon retry. As specified in the preceding section, these errors do not, in themselves, constitute a disk failure. If you are concerned about recurring soft errors for a particular disk, you can use a command such as the one below, to explicitly fail the disk.

```
server# mfs repair -F -d /dev/rdisk/c1t3d0s2 -v
```

Following completion of this command, you can then replace the now-failed disk, following the procedure described in Section 8.3, “Replacing a Failed Disk.”

## 8.2.4 mkmfs

The `mkmfs` utility creates an MFS. This command is automatically invoked by the `pkgadd` installation process. You only need to invoke this utility if you reconfigure the data and parity disks used by the MFS. The `mkmfs` utility is analogous to the Unix `mkfs` (1M) utility.



---

**Caution** – `mkmfs` disables access to all MFS files in existence prior to your running the command. After running `mkmfs`, you must re-run `mfs record` to re-create any media files that had previously existed.

---

`mkmfs` usage is as follows:

```
/opt/SUNWsms/mfs/bin/mkmfs [ -f <config file> ]
```



Following invocation without the `-f` option, `mkmfs` prompts you for the information required to build an MFS. For example:

```
server% /opt/SUNWsms/mfs/bin/mkmfs
FS path name: /var/opt/SUNWsms
Total number of drives: 12
Number of parity drives: 2
Drive name: /dev/rdisk/clt1d0s2
Drive name: /dev/rdisk/clt2d0s2
Drive name: /dev/rdisk/clt3d0s2
Drive name: /dev/rdisk/clt4d0s2
Drive name: /dev/rdisk/clt5d0s2
Drive name: /dev/rdisk/c2t1d0s2
Drive name: /dev/rdisk/c2t2d0s2
Drive name: /dev/rdisk/c2t3d0s2
Drive name: /dev/rdisk/c2t4d0s2
Drive name: /dev/rdisk/c2t5d0s2
Drive name: /dev/rdisk/clt6d0s2
Drive name: /dev/rdisk/c2t6d0s2
<Ctrl-D to exit>
```

You specify the parity drive(s) after you have entered all data drives. The number of parity drive entries must correspond to the number you entered at the “Number of parity drives:” prompt. In the example above, there is one parity drive specified, so `mkmfs` assumes the last entry, `/dev/rdisk/clt6d0s2`, is that single parity drive.

When you use the `-f` option, you must supply the pathname of a file that contains the information required to build an MFS. Shown below is an example disk configuration file that contains the information entered manually in the previous example.

```
/var/opt/SUNWsms/MFS
12
2
/dev/rdisk/clt1d0s2
/dev/rdisk/clt2d0s2
/dev/rdisk/clt3d0s2
/dev/rdisk/clt4d0s2
/dev/rdisk/clt5d0s2
/dev/rdisk/c2t1d0s2
/dev/rdisk/c2t2d0s2
/dev/rdisk/c2t3d0s2
/dev/rdisk/c2t4d0s2
/dev/rdisk/c2t5d0s2
/dev/rdisk/clt6d0s2
/dev/rdisk/c2t6d0s2
```

The meta-file for the MFS is stored in `/var/opt/SUNWsms/MFS`. Use the Solaris `format` command to obtain the SCSI target numbers used as input for `mkmfss`. The preceding example assumes you use slice 2 (`s2`) to stand for the entire disk. Slice 2 is one of the default slice assignments in effect when you install Solaris.

---

## 8.3 Replacing a Failed Disk

The Sun MediaCenter server's parity mechanism provides that, in the event of a single disk failure, the server continues to deliver streams with no user-perceptible loss in performance. If you do experience a single failed disk, you can replace the failed disk with a new disk without interrupting the server's delivery of video bit streams.

---

**Note** – For Sun MediaCenter servers equipped with SPARCstorage MultiPack enclosures, remove the side cover on the enclosure and check the lights on the disk modules. When the server is delivering content, you should see the lights for all of the data disks blink, while the light for the parity disk (the bottom disk on the right as you face the left side of the enclosure) is lit but unblinking. If you see the light for a parity disk blinking, it indicates a failed data disk for that enclosure. As content is being delivered, you should be able to spot the data disk whose light is not blinking.

---

For the purposes of the Sun MediaCenter server, a disk is not failed until you receive a console message such as the following:

```
WARNING: data disk 0x80009a (32,154) FAILED
```

The most effective method for detecting MFS disk failure is to use the WebAdmin tool to monitor the server. See Section 8.1, "Monitoring MFS Disks," for a discussion of ways you can detect disk failures.

If you receive a "disk failed" message, use the `mfs repair` command, with the options described below, to return your MFS to the state it was in prior to disk failure. The `mfs repair` command is described in detail in the *Sun MediaCenter Server Service Manual*.

1. After receiving a "disk failed" message, as shown above, enter the following command:

```
server# mfs repair -s  
drive /dev/rdisk/clt4d0s2 (dev 0x80009a) is FAILED
```

In the output above, make note of the SCSI disk address (similar to the number `c1t4d0s2`). Chapter 3 of the *Sun MediaCenter UltraSPARC Hardware Manual* has instructions for mapping SCSI disk addresses to physical disks.

**2. Open the appropriate Multipack enclosure, locate the failed disk, and remove and replace it.**

Documentation for your disk subsystem tells you how to replace a disk. There is no need to halt the server or reduce its load while replacing the failed disk.



---

**Caution** – It is only at a point after you have received the “disk failed” message described above that you can safely remove and replace a failed disk. Do not confuse this definitive message with, for example, “soft errors,” such as SCSI read errors. If you remove a disk that is not failed while the server is in operation, you can cause extreme disruption to the server, possibly including a hang or a panic.

---

**3. Following disk replacement, enter the following `mfs repair` command to restore data from the parity disk to the new disk:**

```
server# mfs repair -r -d /dev/rdisk/c1t4d0s2 -v
```

The address `c1t4d0s2` is an example, consistent with the preceding example commands.

**4. When the preceding command completes, enter the following command to return the server to normal playback mode, phasing out parity mode.**

```
server# mfs repair -p -d /dev/rdisk/c1t4d0s2 -v
```

Upon successful return of the preceding command, the server is in normal playback mode and you can, once again, withstand a single disk failure.

You can combine the `-r` and `-p` options, shown in the previous two example commands. For example:

```
server# mfs repair -rp -d /dev/rdisk/c1t4d0s2 -v
```

With a command such as the preceding, you can combine repair and return-to-normal-playback in a single command.

The procedure to replace and restore a parity disk is the same as for a data disk. A parity disk failure would be discovered while loading content. At the point of disk failure, the loading operation ceases. As with a data disk, you use the `-r` option to `mfs repair` to restore the parity disk.

---

**Note** – The status of a disk as failed does not persist across a reboot. If you experience a disk failure, then reboot, you must explicitly fail the disk, with a command such as:

```
server# mfs repair -F -d /dev/rdsk/c1t4d0s2 -v
```

Following this command, follow the procedure specified above to replace the failed disk.

---

If you experience multiple, overlapping disk failures, you must restore the entire contents of the MFS from a backup source, using `smc_tar`; `smc_copy`; or `ftp`, in conjunction with the Sun MediaCenter FTP daemon. You might need to use the `smc_tar`'s `w` option, which allows you to overwrite existing content.

---

**Note** – Two overlapping disk failures result in a system panic.

---

If you have to remake the MFS, use the following command:

```
# mkmfs -f /etc/opt/SUNWsms/mfs.file
```

The `mkmfs` command is described in the *Sun MediaCenter Server Service Manual*.

## Access Control Lists

---

You restrict Sun MediaCenter server operations to specific users by creating access control lists (ACLs). You can create ACLs for the following:

- **Server:** Allows programs to list players, titles, or states and/or create or delete players.
- **Players:** Allows programs to obtain play information, control play, and/or set player access on a server.
- **Titles:** Allows users or programs to obtain title information, write or append title data, and/or delete or rename titles or change the title ACL.
- **Sessions:** Allows users or programs to access a Content Manager (CM) session.

The server and player ACLs are associated with the Media Stream Manager (MSM); title and session ACLs are associated with the CM.

---

### 9.1 ACL Files

Server, title, and session ACLs support the use of files that you create; by default, none of these files exists. There is no file associated with player ACLs, although you can programmatically obtain information about or change player access.

The following ACL files are stored in `/etc/opt/SUNWsms`:

- `ServerAcl` for the server
- `ServerTitleAcl` for titles
- `ServerCMSessionAcl` for sessions

As mentioned previously, these files do not exist by default. In the absence of `ServerAcl`, all users have full permissions for a Sun MediaCenter server. In the absence of `ServerTitleAcl` and `ServerCMSessionAcl`, titles and sessions, respectively, have full permissions for their creators and no permissions for anyone else.

---

## 9.2 Permissions

All files used for ACLs support the following syntax:

`<user>:<permission>`

For all ACL files, standard Solaris lookup mechanisms are used for user name lookup. In support of unknown UIDs, all files support:

`nobody:<permission>`

Each type of ACL file allows different permissions; the permissions are described in the following sections in this chapter. For all ACLs, permissions are independent of one another. That is, there is not a hierarchy of permissions in which a "higher" permission includes "lower" permissions.

---

## 9.3 Server ACL

Access to the Sun MediaCenter server is specified in the file `/etc/opt/SUNWsms/ServerAcl`. TABLE 9-1 shows the permissions you can specify:

TABLE 9-1 Server ACL Permissions

Permission	Access Level	Program can call...
r	read	msmPlayerLookup() (to look up, not to create, a player) msmPlayerList() msmPlayerGetPlaylist() msmPlayerGetConnect() msmPlayerGetPlayStatus() msmTitleGetStatus() msmTitleList()
a	admin	msmPlayerLookup() (to create, not to look up, a player) msmPlayerDelete()

By default, there is no `ServerAcl` file, which means that all users have read and admin permission for a Sun MediaCenter server. `ServerAcl` supports a wildcard (\*) to stand for all users, so that you can specify:

`*:r`

...with the result that all users have read permission. For example, with only read permission for a Sun MediaCenter server, a user can look up titles on the server, but cannot play a stream.

Each instance of the `ServerAcl` file must contain a line that indicates the version number of the file; this line must be the first uncommented line in the file. The format of the version line is:

```
Version <number>
```

If the version line is missing or incorrectly formatted, or has an unsupported version number, the MSM produces warning messages but continues to parse `ServerAcl`.

---

## 9.4 Player ACL

There is no file for player ACLs. When you create a player (using `msmPlayerLookup()`), you, the creator, have full permissions—read, control, and admin—for that player. You change permissions by assigning one of a set of access levels to the `MsmAccessList` structure. Permissions are assigned to a specific user. TABLE 9-2 shows the permissions that you can specify:

**TABLE 9-2** Player ACL Permissions

Permission	Access Level	Program can call...
r	read	<code>msmPlayerGetAccess()</code> <code>msmPlayerGetPersistence()</code> <code>msmPlayerGetPlaylist()</code> <code>msmPlayerGetPlayStatus()</code>
c	control	<code>msmPlayerPlay()</code> <code>msmPlayerPause()</code> <code>msmPlayerResume()</code>
a	admin	<code>msmPlayerDelete()</code> <code>msmPlayerSetAccess()</code> <code>msmPlayerSetPersistence()</code> <code>msmPlayerSetPlaylist()</code> <code>msmPlayerSetConnect()</code>

Following creation of a player, the MSM consults the player's ACL before checking `ServerAcl`.

---

## 9.5 Title ACL

Each content title has an ACL which is accessible either programmatically through the `cmGetAcl()` function or by the `smc_gettacl` utility. To create or modify permissions, use the `cmSetAcl()` function or the `smc_settacl` command.

TABLE 9-3 shows the permissions that you can specify:

**TABLE 9-3** Title ACL Permissions

Permission	Access Level	User can run...	Program can call...
r	read	<code>smc_ls</code> to obtain listing of title. <code>smc_copy</code> , <code>smc_tar</code> , and <code>ftp</code> to copy existing content (as in a backup). <code>smc_gettacl</code> to read title ACL.	<code>cmOpen()</code> (for read, not append) <code>cmGetAcl()</code> <code>cmContentList()</code> <code>cmTitleStat()</code>
w	write	<code>smc_tar</code> and <code>ftp</code> to create new content.	<code>cmOpen()</code> (for append, not read) <code>cmCreate()</code>
a	admin	<code>smc_copy</code> to rename a file during a copy operation. <code>smc_settacl</code> to set a title ACL.	<code>cmSetAcl()</code> <code>cmRename()</code> <code>cmDelete()</code>

When you create a title (by using `cmCreate()` from the CM Client Lib API or by using `smc_copy`, `smc_tar`, or `ftp` to move content onto a server), you, the creator, have read, write, and admin permissions for that title. If no other restrictions are imposed by the `ServerAcl` file, a user other than the title's creator cannot copy, delete, or append to a title, unless the creator uses `smc_settacl` or `cmSetAcl()` to open up access to the title.

---

## 9.6 Session ACL

A CM session is a logical relationship between the Sun MediaCenter server and another machine used for copying, appending, and deleting titles and obtaining/changing title ACLs.

Each CM session has an ACL associated with it.



A session ACL has a single type of permission.

**TABLE 9-4** Session ACL Permissions

Permission	Access Level	User can run...	Program can call...
a	admin	CmAdmin to abort a session	cmSetSessionAcl() cmClientClose() cmRpcSetTimeout() cmOpen() cmCreate() cmSessionAbort() cmSessionClose()

A programmer who invokes `cmClientCreate()` to create a session obtains admin access to that session. A user who invokes `smc_copy`, `smc_tar`, or `ftp`, to move, delete, or rename content receives admin access to the CM session used for that operation.



## Troubleshooting

---

This chapter describes tools you can use to track down problems you might have with your Sun MediaCenter server.

The most useful tool you can have for modeling your server's behavior and for detection of problems is SunNet Manager or another SNMP-conformant management platform that allows you to read values collected by the Sun MediaCenter server's SNMP agent in its Management Information Base (MIB). See Appendix A for a description of this MIB.

Another useful diagnostic tool is `snoop`, which is shipped with Solaris.

See the documentation for your disk subsystem for instructions on replacing a disk within the subsystem, should that step be necessary.

---

### 10.1 Symptoms and Possible Solutions/ Workarounds

#### *Server watchdog resets on start of video stream playback*

Possible cause is that the SCSI cables are not connected to correct port or not connected properly. At the monitor prompt, use `probe-scsi-all` to confirm that the server can "see" all disks in the MFS.

Replace unresponsive disk(s) and reload content. As an alternative to reloading content, you can contact your Sun authorized service provider to have content repaired from the parity disk.

### *Underperforming stream output*

Possible cause is network congestion. Use a network analyzer or your switch vendor's diagnostics to analyze network performance.

Note that your server is intended only for the storage and delivery of multimedia streams. Any processes not associated with this purpose have the potential to disrupt server output.

### *No video displays on client*

Possible causes: Disk cables and/or network links not connected. If using ATM, ATM switch not configured correctly. Content is stored at a bit rate different from that at which it was encoded.

If you have SunNet Manager, do a quick dump of the `sms-vod>active` attribute to confirm that there are active streams. If you do not have SunNet Manager, use `snoop`, `atmstat`, and/or switch diagnostics to check on streams. If there are active streams, confirm that the bit rate you are using to play the stream is correct.

Note that on a correctly running server, the following processes should be running:

root	4	0	1	Nov 13 ?	0:00 vod_hires_timer
root	231	1	0	Nov 13 ?	0:00 vod_xor
root	232	1	0	Nov 13 ?	0:00 vod_xor
root	233	1	80	Nov 13 ?	0:13 vod_sched

If you are using an application based on the Media Stream Manager Client API, the following process must be running:

root	241	1	80	Nov 13 ?	2:18 /opt/SUNWsms/msm/bin/MsmServer
------	-----	---	----	----------	-------------------------------------

If any of the preceding processes is not present in `ps` output, you might have to reboot the server.

Another possible cause of this problem, for Fast Ethernet destinations, is that the IP address and/or Ethernet address of the destination client are not recorded in a name service accessible to the server or not in the server's local `/etc/hosts` and `/etc/ethers` files.

#### *Poor video quality*

Possible causes: Content is stored at a bit rate different from that at which it was encoded. Content might be poorly encoded. The destination client might not be able to keep up with the video stream.

Check the bit rate of the content. The bit rate should be roughly  $\text{filesize} * 8 / \text{playtime}$ , where *playtime* is the duration of a title, in seconds. Try another title, preferably one encoded at a lower bit rate. Try a different destination client.

#### *Corrupt disk labels*

You cannot complete the boot process because of corrupt disk labels.

Possible cause: shutting down a server which is the midst of delivering streams.

Reboot single-user and relabel affected disks with `format`.

#### *Kernel panic with SAHI queue threshold messages*

Likely cause: ATM cable is disconnected.

#### *Disk failure*

If you have a single bad disk, use the procedure specified in Chapter 8 “Administering MFS Disks,” to replace the disk and restore data to the new disk. You do not have to interrupt the operation of the server to do this. If you have multiple simultaneous data disk failures, you must, following replacement, use `smc_tar` or `ftp` to restore original content.

#### *Inconsistent video performance*

Possible cause: “soft” disk errors.

Check `/var/adm/messages` for console messages indicating non-fatal disk errors. If you have SunNet Manager or other SNMP-conformant manager, check the `diskErrorTable` structure on the server.

If a single disk is displaying frequent non-fatal errors, have your Sun Sales Engineer replace the disk and restore content from the parity disk to the new disk. There is no need to power down the server to do this. Chapter 8 “Administering MFS Disks,” describes a procedure for “hot” disk replacement.

On a Sun MediaCenter server equipped with multiple Fast Ethernet interfaces, this could be caused by not having all (or both) of the Fast Ethernet links connected, if load sharing has been specified for those interfaces. When load sharing is in effect, the Sun MediaCenter software performs load balancing across multiple Fast Ethernet interfaces, treating the multiple interfaces like a single, logical output interface. Failure to connect all Fast Ethernet links results in difficult-to-diagnose stream delivery failures. This is not a requirement on a server with multiple ATM interfaces. Clients can address ATM interfaces individually.

*Unable to create title on the MFS.*

Likely cause: A filled MFS or a disk failure. In this situation, the Sun MediaCenter software removes all files associated with the title.

Use `mfs df`, documented in the *Sun MediaCenter Server Service Manual*, to determine free space in the MFS. Use `mfs repair -s`, documented in the same *Service Manual*, to determine whether a disk has failed.

If you use SunNet Manager, see if you have received the `Disk_Failure` trap. Follow the procedure in Chapter 8 “Administering MFS Disks,” to replace the defective disk.

---

## 10.2 Error Messages

The numbers that appear in each error message listed below are examples only.

### 10.2.1 Server Console Messages

WARNING: data disk 0x80009a (32,154) FAILED

Cause: an MFS data disk has failed. The server switches into parity mode.

Action required: Replace the failed disk, per the procedure described in Chapter 8.

WARNING: vod\_add\_stream: failed ba@466,1,8060@

Cause: Attempt to add a new video stream, which would exceed the output network bandwidth supported by the Sun MediaCenter server.

Action required: Discontinue attempts to add more streams until one or more streams is finished.

WARNING: stream id 96 (ba@325,0,8060@): Deadline Missed

Cause: A disk read operation cannot keep up with the Sun MediaCenter server. Message can be accompanied by short (less than one second) interruptions in the video for affected streams.

Action required: Occasional missed deadlines are not a concern. If you have several in an hour and these are accompanied by interruptions in video display, contact your Sun service representative.

WARNING: stream id 107 (ba@426,3,80@) xmit error: Q size 33

Cause: Low-level software has detected a transmission error.

Action required: None. Most often, higher level software can recover from such an error. However, if you receive such messages continually or if such messages are accompanied by degraded video output, contact your Sun service representative.

NOTICE: ba0: qsize OK

Cause: This is a Sun MediaCenter driver diagnostic message.

Action required: None. It is useful to trained technical personnel, particularly if it occurs in combination with other messages listed here.

NOTICE: ba3: qsize 9 above hiwat

Cause: This is a Sun MediaCenter driver diagnostic message.

Action required: None. It is useful to trained technical personnel, particularly if it occurs in combination with other messages listed here.

## 10.2.2 MFS Error Messages

The `mfs perror` utility displays information about a specific MFS error number.

`perror` usage is as follows:

```
mfs perror error_number
```

The single parameter is the error number returned by MFS when it attempts to execute an MFS utility. For example, an attempt to use `mfs create` to create a file in the MFS may return the following:

```
Can't create title 4260348
```

Use `mfs perror` to obtain further information about the error:

```
server# mfs perror 4260348
mfs_bmap_alloc: Unable to allocate mfs data blocks: Insufficient space on disk
```





## Setting up the SNMP Agent

---

<i>The Sun MediaCenter Server MIB</i>	<i>page A-2</i>
<i>Enabling Management by Solstice Domain Manager</i>	<i>page A-6</i>
<i>Enabling Trap Handling</i>	<i>page A-9</i>

The Sun MediaCenter software is shipped with its own SNMP agent, which is part of the `SUNWsmsnm` package. This agent is based on the SNMP agent shipped with the Solstice Domain Manager (SDM); the agent has been extended to support the Sun MediaCenter server. This appendix tells you how to configure the SNMP agent, and how to access that agent using Solstice Domain Manager.

---

**Note** – Because you need to provide site-specific information when installing the SNMP agent, the `SUNWsmsnm` package is *not* preinstalled on the Sun MediaCenter server. Therefore, if you want to use the SNMP agent to manage the Sun MediaCenter, you must install the `SUNWsmsnm` package from the “Sun MediaCenter Software 2.1 Recovery” CD-ROM. To do this, access the CD-ROM and run the `swmtool` or `pkgadd` program, then select the `SUNWsmsnm` package for installation. (These steps are described in detail in Chapter 4 in the *SunMediaCenter Server Installation and Configuration Manual*. The prompts that you will need to respond to for `SUNWsmsnm` installation are also described in that manual.)

---

The SNMP agent described here is compatible with any SNMP-capable management product. You are not limited to the use of Solstice Domain Manager in managing the Sun MediaCenter server.

---

**Note** – A caveat on the use of the SNMP agent shipped with the Sun MediaCenter server: If you use SNMP to manage the Sun MediaCenter software, you cannot also run an SNMP agent to manage other components of your system (for example, the ATM interface). The SNMP agent reserves port 161, making that port unavailable to other SNMP agents.

---

The SNMP agent is implemented in a daemon, `snmpd`, and is installed on the Sun MediaCenter server in the course of running `pkgadd` to add the `SUNWsmsnm` package. The package-install script adds a `snmpd` startup script to `/etc/init.d`. Beyond running `pkgadd`, no action is required on your part to install or start `snmpd`.

---

## A.1 The Sun MediaCenter Server MIB

The Sun MediaCenter Server Management Information Base (MIB) is defined in the file `sms_vod.mib`, stored by default in `/opt/SUNWsms/agents`. A MIB contains attributes that collectively describe a network object. `sms_vod.mib` describes a Sun MediaCenter server. MIB attributes are readable—and in some cases settable—by a program that conforms to the Simple Network Management Protocol (SNMP). All of the attributes in `sms_vod.mib` are read-only, meaning you cannot use SNMP to change their values.

MIB attributes are categorized by groups and tables. TABLE A-1 lists the groups and tables in `sms_vod.mib`.

**TABLE A-1** Sun MediaCenter Server MIB Definitions

Group or Table	
Attribute Name	Description
active	
<code>activeNumStreams</code>	Number of active streams
<code>activeStreamBandwidth</code>	Collective network bandwidth of all active streams, in bytes/sec.
admission	
<code>admMaxStreams</code>	Maximum number of streams allowed. This a fixed value.
<code>admMaxBandwidth</code>	Maximum bandwidth allowed for streams, in bytes/sec.
<code>admNumAdmitted</code>	Number of successful admissions.
<code>admNumFailed</code>	Number of failed admissions.

**TABLE A-1** Sun MediaCenter Server MIB Definitions

Group or Table	
Attribute Name	Description
admTotalDelay	Total delay for the admission of all streams since last reboot, in milliseconds.
admAverageDelay	Average admission delay, in milliseconds.
bitPump	
bitPumpStreams	Current number of bit pump streams.
bitPumpBandwidth	Bandwidth of the bit pump, in bits/sec. This is a fixed value.
bitPumpMissedDeadlines	Number of missed deadlines.
bitPumpStreamTable	
bitPumpStreamId	Number that identifies a specific bit pump stream.
bitPumpStreamDrift	Number of milliseconds of drift for a given stream.
bitPumpStreamMTU	Size of the UDP payload for each transmitted Ethernet packet; size of the AAL5 payload for each transmitted ATM packet.
bitPumpStreamLate	Number of times a stream is late.
bitPumpNumDiskIOs	Number of disk I/Os for a stream.
bitPumpNumDeviceIOs	Number of device I/Os for a stream.
diskError	
diskErrors	Number of disk errors.
diskErrorTable	
diskErrorId	Number identifying the error.
diskErrorDevice	Name of the disk having the error.
diskErrorTimes	Time the error occurred.
diskErrorType	Description of the error as returned by the operating system.
diskErrorBlockNumber	Block number of the errant block.
diskErrorNumBlocks	Number of errant blocks.
diskLate	
diskLateEvents	Number of late disk events.
diskLateTable	
diskLateId	Instance number of the late event.
diskLateDevice	Name of the late device.

**TABLE A-1** Sun MediaCenter Server MIB Definitions

Group or Table	
Attribute Name	Description
diskLateTime	Time at which the late event was detected, in microseconds.
diskLateDuration	Duration of lateness, in milliseconds.
diskTable	
diskDeviceId	Name of the disk.
diskNumIOoperations	Number of I/O operations for a disk.
diskNumLate	Number of times a disk was late.
diskTotalLateness	Total time late, in seconds.
diskNumErrors	Total number of errors for a disk.
diskType	"data" or "parity".
failure	
failNumOfDisks	Total number of failed disks.
failNumRecovering	Number of disks that have failed and are undergoing recovery through the parity mechanism. Should be no more than 1.
failDevice	Name of the last failed device.
failLastTime	Time of the last failed device.
failNumXOR	Number of parity recoveries (XOR).
online	
onlineTotalBandwidth	Total usable bandwidth for online loading.
onlineMaxTransfer	Overall maximum transfer size of online I/O.
onlineAvailableBandwidth	For a given instant, currently available bandwidth per time unit (see bitPump:bitPumpTimePeriod).
onlineMaxAvailTransferSize	Maximum, currently available transfer size for online I/O.
onlineNumStarted	Number of online loading requests started.
onlineNumFinished	Number of online loading requests finished.
vodsystem	
sysDiskBandwidth	Bandwidth of each disk (without considering seek overhead).
sysTotalDiskBandwidth	Total bandwidth of all data disks.
sysNumberOfDisks	Number of data disks.
sysNumberOfParityDisks	Number of parity disks.

**TABLE A-1** Sun MediaCenter Server MIB Definitions

Group or Table	
Attribute Name	Description
sysZonesPerDisk	Number of zones per disk. A zone is a region of a disk used for the storage of streams
sysTimePeriodLength	A fixed interval used by VOD scheduler for MFS operations.
sysNumberOfBuffers	Number of configured buffers.
sysBufferSize	Size of each buffer.
sysBufferMax	Maximum number of buffers used at one time.
sysBufferUnavail	Number of times buffers not available.
sysBuffersInUse	A snapshot of the number of busy buffers.
sysCyclesCompleted	Time period cycles completed.

The SMC MIB attributes can be categorized a number of different ways. One way is to divide the attributes between those that have fixed values—fixed at compile time or when the server software completes booting—and those that change in response to server use. Examples of the former category are listed in TABLE A-2.

**TABLE A-2** Fixed-value Attributes

Attribute Name	Description
admMaxStreams	Maximum number of streams allowed.
admMaxBandwidth	Maximum bandwidth allowed for streams, in bytes/sec.
bitPumpBandwidth	Bandwidth of the bit pump, in bits/sec.
sysDiskBandwidth	Bandwidth of each disk (without regard to seek overhead).
sysTotalDiskBandwidth	Total bandwidth of all data disks.
sysNumberOfDisks	Number of data disks.
sysNumberOfParityDisks	Number of parity disks.
sysZonesPerDisk	Number of zones per disk.
sysTimePeriodLength	An interval used by VOD scheduler for MFS operations.
sysNumberOfBuffers	Number of configured buffers.
sysBufferSize	Size of each buffer.

The remaining attributes, those not in TABLE A-2, are counters, sizes, identifiers, and times (both durations and instants) that change according to conditions on the server. These attributes can be categorized as follows:

- Streams and stream admission  
The active and admission groups.
- Bit pump  
The bitPump group and the bitPumpStreamTable.
- Disk I/O  
The diskError group, including the diskErrorTable; the diskLate group, including the diskLateTable; the diskTable; the failure group; the system group.
- Online loading  
The online group.

Attributes that are associated with tables are designed so that if an attribute indicates a certain number of errors, you can retrieve the related table to find out details about the error conditions. For example, if you observed a higher-than-normal number for `bitPump:bitPumpMissedDeadlines`, you could retrieve the `bitPumpStreamTable` to find out details about specific bit pump streams.

---

## A.2 Enabling Management by Solstice Domain Manager

The following procedure is for enabling management of the Sun MediaCenter server's SNMP agent in Solstice Domain Manager. You perform these steps on the SNM Console machine. If you use a management platform other than Solstice Domain Manager, consult the documentation for that product.

1. **Become root.**
2. **Copy** `/opt/SUNWsms/agents/sms1000.icon` **and** `sms1000.iconmask` **to** `$SNMHOME/icons`.

**3. Edit `$(SNMHOME)/struct/elements.schema`, to add lines for the Sun MediaCenter server.**

Use the lines for the SPARCcenter 1000 (`sc1000`) as a basis for the new lines for the Sun MediaCenter server (`sms1000`). For example:

```
record component.sc1000 (                # SC1000
    string[64]      Name
    string[40]      IP_Address
    string[40]      User
    string[40]      Location
    string[80]      Description
    string[40]      SNMP_RdCommunity
    string[40]      SNMP_WrCommunity
    string[64]      SNMP_Vendor_Proxy
    int             SNMP_Timeout
)
Following lines copied from the previous lines
record component.sms1000 (                # SC1000
    string[64]      Name
    string[40]      IP_Address
    string[40]      User
    string[40]      Location
    string[80]      Description
    string[40]      SNMP_RdCommunity
    string[40]      SNMP_WrCommunity
    string[64]      SNMP_Vendor_Proxy
    int             SNMP_Timeout
)
```

The remaining additions to `elements.schema` are as follows:

***Under “instance elementCommand”:***

```
(component.sc1000      "Rlogin..."  "$(SNMHOME)/bin/snm_cmdtool rlogin %Name")
(component.sc1000      "Telnet..."  "$(SNMHOME)/bin/snm_cmdtool telnet %Name")
(component.sms1000     "Rlogin..."  "$(SNMHOME)/bin/snm_cmdtool rlogin %Name")
(component.sms1000     "Telnet..."  "$(SNMHOME)/bin/snm_cmdtool telnet %Name")
```

***Under “Glyphs for Components”:***

```
( component.sc1000      sc1000.icon)
( component.sms1000     sms1000.icon)
```

**4. Copy `sms_vod.mib` to `$(SNMHOME)/agents`.**

**5. Run: `$(SNMHOME)/bin/mib2schema sms_vod.mib`**

This command creates `sms_vod.mib.schema` and `sms_vod.mib.oid`.

**6. Run: `build_oid` (with no arguments).**

This program maps names in Sun MediaCenter server MIB to new OIDs.

**7. (Optional) Copy your `$SNMHOME/discover.conf` file to `/etc/opt/SUNWconn/snm`. In the MAPPINGS section of this copy, add the line:**

<code>media</code>	<code>component.sms1000</code>
--------------------	--------------------------------

In addition to modifying `discover.conf`, set the environment variable `SNMDISCOVERMAP` to the directory where `discover.conf` is located.

This step enables the Discover tool to find a Sun MediaCenter server and to identify that machine correctly as a Sun MediaCenter server. This is a convenience to you but is not required for the effective management of a Sun MediaCenter server.

**8. Save your Management Database then restart the Console, using `snm -i`.**

The `-i` option recreates your SNM database, including the new, video server schema file. Take the precautions described in the SNM documentation before entering this command.

---

**Caution** – It is critical that you save you save your Management Database before invoking `snm` with the `-i` option.

---

**9. In Console window, obtain the Properties sheet for a Sun MediaCenter server, available through the server icon's pulldown menu.**

This step presumes you have run Discover or manually added the Sun MediaCenter server to your database.

**10. In the Properties sheet, select the media server agent, `sms_vod`, and select Apply.**

**11. In the server icon's pulldown menu, select Change Type. In the Change Type window, select `sms1000`.**

In the Console window, the icon for the selected machine changes to an `sms1000` icon.

You now can use SNM to manage one or more Sun MediaCenter servers.



As an example of a the SNMP agent's capability, the following is a Quick Dump report for the `bitPump` attribute on a machine `redoubt`:

```
Wed May 24 15:12:15 1995 [ redoubt ] : Quick Dump: sms-vod.bitPump
bitPumpStreams=2
bitPumpBandwidth=8003840
bitPumpTimePeriod=5405:07:40.72
bitPumpMissedDeadlines=0
```

---

## A.3 Enabling Trap Handling

In addition to the six standard SNMP traps, the Sun MediaCenter server's SNMP agent supports three enterprise-specific traps, `Admission_Failed`, `Missed_Deadline`, and `Disk_Failure`. When the bit pump misses a deadline, the server fails to admit a title, or a disk failure occurs, the agent sends enterprise-specific trap 1, 2, or 3, respectively.

Perform the following tasks to enable trap handling in Solstice Domain Manager.

SNM uses the `snmp.traps` file, stored by default in `/var/opt/SUNWconn/snm`, to establish mapping between trap numbers and descriptive names for those traps. To map the enterprise-specific trap numbers (1, 2, and 3, as mentioned above) to names, on the SNM Console machine, perform one of the following two tasks:

- If you have not made use of the `snmp.traps` file to this point, save the file to a file of a different name. Then, from the Sun MediaCenter server, copy the `snmp.traps` file in `/opt/SUNWsms/agents` to `/var/opt/SUNWconn/snm` on the SNM Console machine.
- If you have already modified `snmp.traps` on the Console machine and want to preserve its contents, append the contents of `snmp.traps` on the Sun MediaCenter server to its counterpart on the Console machine.

See the *Solstice Domain Manager Administration Manual* for details on the `snmp.traps` file.

The trap explanations are:

### 1 Admission\_Failed

The Sun MediaCenter server was unable to admit a title that a user requested for playing. This indicates an exhaustion of resources.

## 2 Missed\_Deadline

The Sun MediaCenter software ensures delivery of media streams at the same rate at which they were recorded. The bit pump's missing of a deadline indicates that the machine was, at least momentarily, unable to keep up with the rate of a title. This indicates a resource conflict on the server.

## 3 Disk\_Failure

The disk subsystem sustained a failure. The failure can occur at when you are attempting to load content on the Sun MediaCenter server. More likely the failure occurs when data is being retrieved from the disk subsystem for output. Because of its parity system, the Sun MediaCenter server can recover from a single disk failure.

To enable specified machines to receive traps—both SNMP-standard and enterprise-specific—generated by the Sun MediaCenter server, on the server itself, edit `/etc/opt/SUNWsms/snmpd.conf`. Change:

trap	localhost
------	-----------

to:

trap	<host1> <host2> ...
------	---------------------

Note the syntax for multiple trap recipients: a single space between arguments. The trap arguments can be hostnames or IP addresses.

## Creating a Content Package

---

<i>Content Package</i>	<i>page B-2</i>
<i>The TOC File</i>	<i>page B-3</i>
<i>Simplest Case TOC File</i>	<i>page B-8</i>
<i>TOC File for Large Files</i>	<i>page B-9</i>
<i>Index File Requirements</i>	<i>page B-10</i>
<i>Tape Storage Format</i>	<i>page B-12</i>
<i>Example</i>	<i>page B-14</i>
<i>References</i>	<i>page B-18</i>

This appendix describes a *content package*, which is one way that content can be prepared for storage on a Sun MediaCenter server. Specifically, this document describes:

- the types of files that make up a content package;
- the storage format of those files on tape.

Under most circumstances, you do not have to deal with a content package. The Content Manager (CM), an integral part of the Sun MediaCenter software, extracts from MPEG content—automatically and invisibly—the information the server needs to successfully store and play that content.

The only circumstance in which you must create a content package is when your content is stored on a non-Solaris machine that does not have `ftp` client software.

If this circumstance applies to you or you choose to create a content package, load your content using either `CmTool` (described in Chapter 4) or `smc_copy` or `smc_tar` (described in Chapter 5).

---

## B.1 Content Package

A *content package* is a collection of separate MPEG streams that represent the content of a title (including audio and video) at various playback speeds.

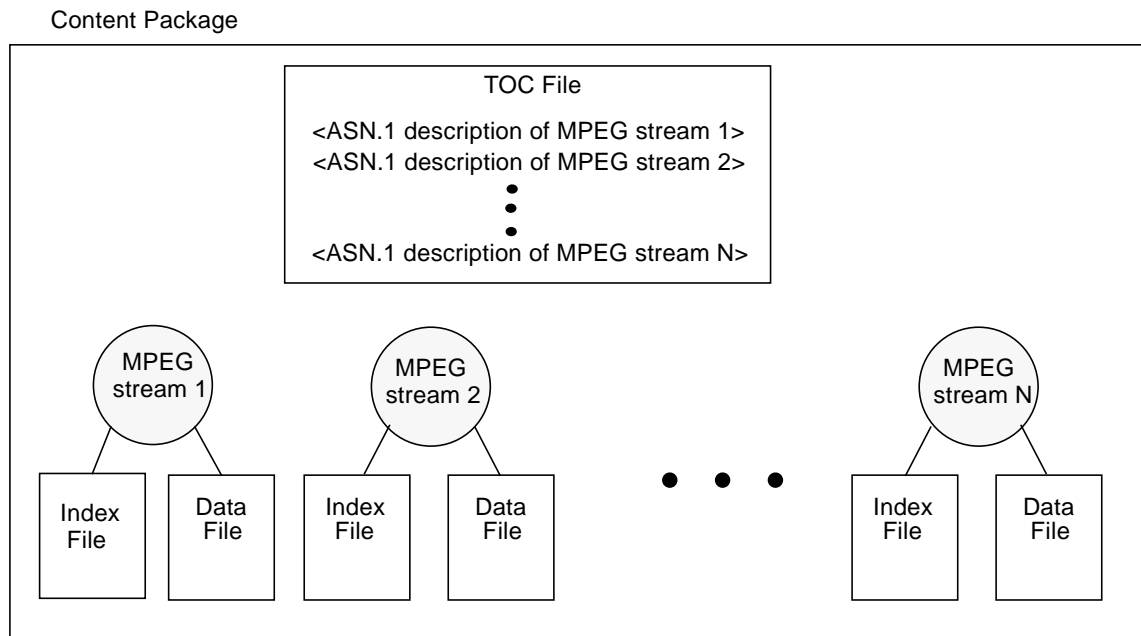
There are two types of files that together describe each of the (potentially) multiple bit streams that make up a content package:

- the data file containing the MPEG-encoded data;
- the index file that contains information on how a specific data file is related to the rest of the content.

In addition to a data file/index file pair for each bit stream, a content package contains a single Table of Contents (TOC) file. The TOC file contains information on each of the bit streams in the content package. The TOC file is covered in detail in the following section.

A content package must contain at least one bit stream, the bit stream representing the movie at normal playback speed. See Section 2.2 “MPEG Encoding Requirements for Trick Play” on page 2-2 for a discussion of the use of additional bit streams to support VCR-like functions, such as fast-forward and rewind.

FIGURE B-1 illustrates the concepts inherent in a content package.



**FIGURE B-1** Structure of a Content Package

---

**Note** – If your content consists of a single bit stream, you do not need an index file. Your content package will consist of only a Table of Contents file and the single bit stream.

---

## B.2 The TOC File

The Table of Contents (TOC) file is an ASCII file that contains:

- a list of all of the bit streams in the content package;
- the individual data and index files that make up these bit streams;
- relevant attributes of these bit streams.

The structure of the TOC file is specified using the Abstract Syntax Notation One (ASN.1). ASN.1 is a standardized way for describing structured information. See Section B.8 “References” on page B-18 for the standards that specify ASN.1.

A TOC file is described with the following structures:

```
content ContentObjects
ContentObjects ::= SEQUENCE
{
    title          GraphicString,
    version         GraphicString,
    format          GraphicsString,
    description     GraphicStringOPTIONAL,
    bitstreams      BitStreamObjects
}
BitStreamObjects ::= SEQUENCE
{
    description     GraphicStringOPTIONAL,
    speed           INTEGER,
    bitrate         INTEGER,
    indexfile       GraphicString,
    indexsize       INTEGER,
    datafile        DataFileObjects
}
DataFileObjects ::= SEQUENCE
{
    segment         INTEGER,
    filepath        GraphicString,
    filesize        INTEGER
}
```

Some rules regarding the encoding of a TOC file:

- TOC files themselves are assignments to the types declared above.
- All INTEGERS are represented as decimal values. These are all 32-bit values, unless other noted in this document.
- All strings are expressed as a single value enclosed by a pair of double quotation marks. No concatenation of strings is allowed.

The following subsections describe the components and properties of a content package.

## B.2.1 ContentObjects

The structure contains the overall information for the content package.

### B.2.1.1 title

The title property is a string that specifies the name of the content. Use the convention:

```
title "<stock symbol> <titlename>" ,
```

For example, if you want to load the content for the title, “Bambi”, fill in the title field as follows:

```
title "SUNW Bambi"
```

The Media Stream Manager replaces the space between *<stock symbol>* and *<titlename>* with an underbar, so that the name of the title displays (to a Media Stream Manager application) as `SUNW_Bambi`.

The convention of using a stock symbol in conjunction with the title name allows the Media Stream Manager to distinguish among titles of like content that originate from different sources.

### B.2.1.2 version

The version property is a string that specifies the version of the content-input format used to generate the content package. The version number of the format specified by this document is `SUNW.00.01`.

### B.2.1.3 format

The format property is a string that specifies the encapsulation used in the content package. Currently supported encapsulation types are:

MPEGTS

MPEG-2 transport packet encapsulation wherein the server transports data in units of 188-byte packets.

#### MPEGTCE

DSS transport format wherein the server transports data in units of 130-byte packets.

#### MPEGPS

MPEG-2 program stream encapsulation wherein the server transports data in 4-byte aligned packet sizes of arbitrary length.

#### MPEG1SYS

MPEG-1 system stream encapsulation wherein the server transports data in 4-byte aligned packet sizes of arbitrary length.

### B.2.1.4 description

The description property is an optional string that allows you to enter a content-specific comment.

## B.2.2 BitStreamObjects

The BitStreamObjects structure describes the bit streams that make up the content package.

### B.2.2.1 description

The description property is an optional string that allows you to enter a bit stream-specific comment.

### B.2.2.2 speed

The speed property specifies the ratio of the playback speed of the bit stream being described, compared to the bit stream at normal playback speed. Normal playback speed is specified by the integer 1000.

The speed value is a positive integer in the forward play direction and a negative integer in the reverse play direction. A bit stream that is five times normal speed in the forward direction has a speed value of 5000. A reverse-play bit stream, at normal speed has a speed value of -1000.



### **B.2.2.3      bitrate**

The bitrate property, expressed in bits/sec., specifies the bit rate at which the MPEG data file was encoded. This number also specifies the rate at which the file is to be played by the server to the network.

### **B.2.2.4      indexfile**

The indexfile property is a string that specifies the pathname to the index file. The index file is an ASCII file that specifies how the bit stream being described is related to the bit stream at normal play speed. The index file is used to switch among bit streams in a content package. It is described in detail in Section B.5 “Index File Requirements” on page B-10.

### **B.2.2.5      indexsize**

The indexsize property is an integer that specifies the number of entries in the index file identified by the indexfile property. An index file entry consists of a time/file-offset pair. See Section B.5 “Index File Requirements” on page B-10 for a description of index file entries.

### **B.2.2.6      datafile**

The datafile property specifies the attributes of the MPEG data for a given bit stream, as described by the DataFileObjects structure.

## **B.2.3      DataFileObjects**

The DataFileObjects structure is a sequence of data files that make up the described bit stream’s MPEG data.

### **B.2.3.1      segment**

The segment property is an integer that specifies the sequence number of the data file in a multi-segment data file. See Section B.4 “TOC File for Large Files” on page B-9 for a description of the use of this parameter.

### B.2.3.2      filepath

The filepath property is a string specifying the pathname of the data file for this segment of the bit stream's data.

### B.2.3.3      filesize

The filesize property is an integer specifying the size of the data file for this segment of the bit stream's data, in bytes. It is a 64-bit integer value.

---

## B.3            Simplest Case TOC File

The following is an example TOC for a title with a single bit stream:

```
content
{
  title "Your title here",
  version "SUNW.00.01",
  format "<MPEG format>",
  description "Enter string here",
  bitstreams
  {
    description "Enter string here",
    speed 1000,
    bitrate <encoding bit rate, in bps>,
    datafile
    {
      segment 1,
      filepath "<path to MPEG file>",
      filesize <size of MPEG file>
    }
  }
}
```

In the TOC file, the critical fields are format, bitrate, and filepath. format and bitrate are critical for the eventual playing of your stream. The filepath field, is critical for the correct operation of `smc_tar`, which you can use to load your content onto the server. `smc_copy` and `ftp`, in conjunction with the Sun MediaCenter FTP daemon, create a TOC file for you.

The fields in the TOC file are described as follows:

`title`

A name of your choosing. After you have stored the title on the server, `smc_ls` returns the contents of this field.

`version`

For versions 1.0 and 1.1 of the Sun MediaCenter server software, this is always `SUNW.00.01`.

`format`

For MPEG-1 content, this is `MPEG1SYS`. For MPEG-2 content, enter `MPEGPS` (MPEG-2 Program Stream) or `MPEGTS` (MPEG-2 Transport Stream), depending on how you encoded your content.

`description`

A string of your choosing. For your own use in describing the entire title. This field is a good place to record all of the PIDs used in your title.

`second description field`

Again, a string of your choosing. This string describes the individual bit stream. You might enter "Normal-play speed".

`speed`

1000 indicates normal-play speed.

`bitrate`

Exact encoding bit rate or multiplexed bit rate where there are audio and video streams.

`filepath`

Path where `smc_tar` will find the MPEG file.

`filesize`

Size of the MPEG file, in bytes.

---

## B.4 TOC File for Large Files

If you have a piece of content that exceeds the Solaris size limit of 2.1 GB, you must divide the MPEG file into smaller files and identify those files as segments in the TOC file for that title. This breaking up of the large file into smaller files occurs at the encoder. Obtain the size, in bytes, of each smaller file and use those values to fill in the `filesize` parameter for each segment in the TOC file.

The following is an example of a TOC file for a multi-part title:

```
content
{
  title "Perils of Technology",
  version "SUNW.00.01",
  format "MPEG1SYS",
  description "History of technology, with Luddite slant",
  bitstreams
  {
    description "Normal play bit stream",
    speed 1000,
    bitrate 1660400,
    datafile
    {
      segment 1,
      filepath "Part1.mpg",
      filesize 284512264
    }
    {
      segment 2,
      filepath "Part2.mpg",
      filesize 132004840
    }
    {
      segment 3,
      filepath "Part3.mpg",
      filesize 260172628
    }
  }
}
```

---

## B.5 Index File Requirements

An index file is an ASCII file that specifies the splice points for one bit stream in a content package. (See Section 2.3 "Splice Points" on page 2-3 for encoding requirements for splice points.) There is a single index file for each bit stream in a content package. The Sun MediaCenter server uses the information in the index file to jump from bit stream to bit stream during trick play.

---

**Note** – For video formatted as MPEG-2 Transport Streams, the CM automatically generates index files for titles that contain trick play streams, at the point when content is loaded onto the server. For other formats, you must generate your own index files.

---

The index file consists of a sequence of splice points. Each splice point is of the following format:

<i>&lt; normal play time &gt;</i> <i>&lt; file offset &gt;</i>
--

Each entry in the index file represents a single splice point. For forward-playing streams, the first entry must be:

0    0
--------

For forward-playing streams, the last pair must be:

<i>&lt; total NPT &gt;</i> <i>&lt; bit stream file size &gt;</i>
--

For reverse-playing streams, the first entry must be:

<i>&lt; total NPT &gt;</i> 0
------------------------------

For reverse-playing streams, the last pair must be:

0 <i>&lt; bit stream file size &gt;</i>
---

## B.5.1 Normal Play Time

The value in the Normal Play Time (NPT) field, expressed in microseconds, specifies the time offset into the program. It is a 64-bit integer value.

The following are requirements and characteristics of NPT in an index file:

- The total NPT should be the same in all index files in a content package.
- NPTs for a forward-playing stream must monotonically increase, from beginning to end of the index file.

- NPTs for a reverse-playing stream must monotonically decrease, from beginning to end of the index file.
- NPT is a time offset into the content as one would view the content at normal speed, not the time at which the pictures are presented (PTS).

## B.5.2 File Offset

The value in the file offset field specifies the byte offset into the MPEG-2 data file corresponding to the start of the splice point. It is a 64-bit integer value.

File offsets must monotonically increase, from beginning to end of the index file.

---

# B.6 Tape Storage Format

Tape storage for the Sun MediaCenter server contains one content package. The tape storage format is the `tar` format. Use the `tar` utility to store the individual files of a content package on a tape. Use the `smc_tar` utility, described in Section 5.3 “The `smc_tar` Utility” on page 5-8, to extract a content package from a tape.

Consult your `tar` documentation for details of the `tar` format and for the syntax of the `tar` command. Unix (including Solaris) users can consult the `tar` (1) man page.

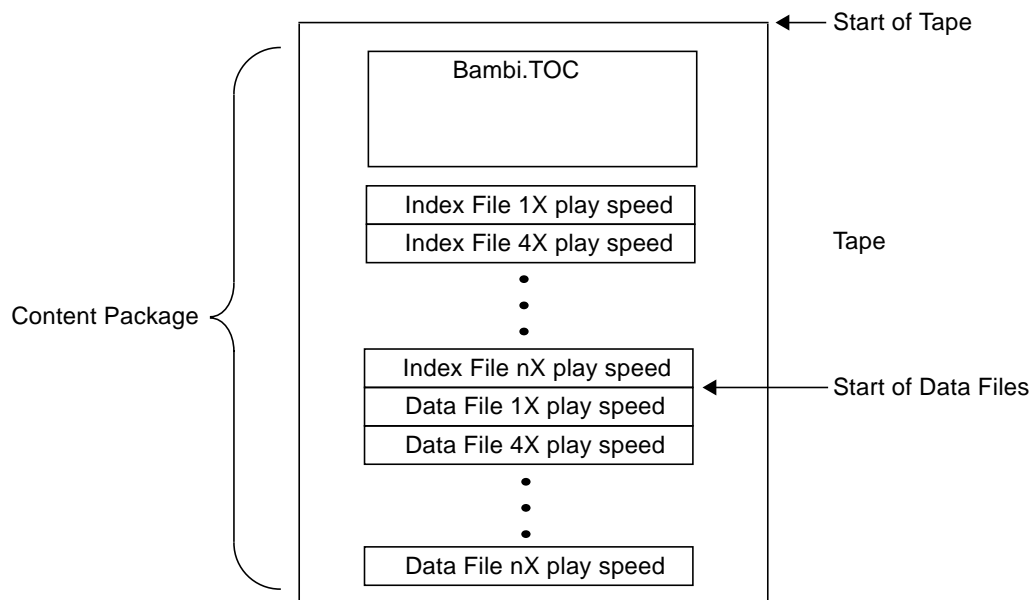
---

**Note** – In the current version, you can have only one content package per tape.

---

## B.6.1 Tar Format

The `tar` format for tape archival maintains an internal table of contents, thus obviating the need for a separate list. The files that are to be extracted from the tape using `tar` are illustrated in FIGURE B-2, below.



**FIGURE B-2** Tape Storage Format

The TOC file must be positioned as the first file in the `tar` tape and must have a file extension of `.TOC`. Except for the file extension, the naming of the TOC file is arbitrary. In FIGURE B-2, the TOC file is named `Bambi.TOC`, to indicate it is the TOC file for the movie “Bambi”. Naming for the data and index files is arbitrary. However, their pathnames must be consistent with those specified in the TOC file.

In FIGURE B-2, note that index files precede data files. This arrangement, while not mandatory, takes advantage of the Sun MediaCenter software’s error checking. The server might find a value in an index file that indicates an error in the storage format of a data file. You can detect and correct the error quickly, rather than, for example, waiting 20 minutes or more to load a 2-hour MPEG title, only to find out there is an error revealed in one of the last index files on the tape.

---

**Note** – If the TOC file contains multiple segments for a `DataFilesObject` (see page B-3), the files corresponding to each of the segments of the `DataFilesObject` must be stored sequentially in the tape archival.

---

The recommended block size of the `tar` format is 500 512-byte blocks, or 256,000 bytes. The default blocking factor for `tar` is 20 blocks, or 10,240 bytes.

## B.6.2 Tapes

The Sun MediaCenter server supports both 4-mm and 8-mm cartridge tapes.

---

## B.7 Example

This section provides an example of a content package—the TOC file and the index and data files for each bit stream—and its storage on a tape using the `tar` format.

The names of the files that make up our example content package are shown in the `tar` output, below:

```
server# tar tf /dev/rmt/0
bambi.TOC
bambi.1x.index
bambi.7x.index
bambi.10x.index
bambi.7nx.index
bambi.10nx.index
bambi.1x.mpeg
bambi.7x.mpeg
bambi.10x.mpeg
bambi.7nx.mpeg
bambi.10nx.mpeg
```

The `tar tf` command above lists the table of contents on a `tar` file or device, `/dev/rmt/0` in this example. The `tar` output indicates that the content package consists of a TOC file, `bambi.TOC`, and five bit streams associated with the content at various speeds. Each bit stream contains an index file (`.index`) and an MPEG data file (`.mpeg`).

In our example, as in FIGURE B-2 on page B-13, the index files for a content package appear immediately after the TOC file on the tape.



The content of the TOC file `bambi.TOC` is shown below. It contains attributes for every bit stream included in the content package for the movie “Bambi”.

**CODE EXAMPLE B-1** Example Content Package

```
content
{
  title "bambi",
  version "SUNW.00.01",
  format "MPEGTCE",
  description "Aftermath of concentration camp survival",
  bitstreams
  {
    description "",
    speed -7000,
    bitrate 2559999,
    indexfile "bambi.n7x.index",
    indexsize 46,
    datafile
    {
      segment 1,
      filepath "bambi.n7x.mpeg",
      filesize 5715190
    }
  }
  {
    description "",
    speed -21000,
    bitrate 2559999,
    indexfile "bambi.n21x.index",
    indexsize 16,
    datafile
    {
      segment 1,
      filepath "bambi.n21x.mpeg",
      filesize 1902680
    }
  }
  {
    description "",
    speed 7000,
    bitrate 2559999,
    indexfile "bambi.7x.index",
    indexsize 46,
    datafile
    {
      segment 1,
      filepath "bambi.7x.mpeg",
```

**CODE EXAMPLE B-1** Example Content Package

```
        filesize 5710380
    }
}
{
    description "",
    speed 21000,
    bitrate 2559999,
    indexfile "bambi.21x.index",
    indexsize 16,
    datafile
    {
        segment 1,
        filepath "bambi.21x.mpeg",
        filesize 1905670
    }
}
{
    description "",
    speed 1000,
    bitrate 5380000,
    indexfile "bambi.1x.index",
    indexsize 250,
    datafile
    {
        segment 1,
        filepath "bambi.1x.mpeg",
        filesize 83554510
    }
}
}
```

The textual content of index files for various speeds are shown below. These examples illustrate the requirements spelled out in Section B.5 "Index File Requirements" on page B-10.

For a file bambi.1x (index file for normal-speed play):

0	0	<i>required NPT and file offset starting values</i>
496978	258310	
993956	595010	
1490935	935610	
1987913	1268670	
2484892	1605500	
.		<i>200+ entries omitted from example</i>
.		
.		
121262750	82085770	
121759728	82422600	
122256707	82759430	
122753685	83096130	
123250664	83432960	
124244621	83554510	<i>Total NPT and bit stream size</i>

For a file bambi.21x (index file for twenty-one times normal speed):

0	0
7765288	109850
15530577	238290
23295866	364910
31061155	490880
38826444	626080
46591732	753870
54357021	876330
62122310	1008800
69887599	1132430
77652888	1262820
85418176	1397370
93183465	1518140
100948754	1648010
108714043	1772290
124244621	1905670

In the preceding example note that the NPT value for the last entry is the same as the NPT entry for normal speed. However, the total bit stream size (file offset value in last entry) is smaller for the trick-play file.

For a file `bambi.n21x` (index file for twenty-one times normal speed, in reverse):

124244621	0
116479333	107510
108714044	239070
100948755	362960
93183466	498550
85418177	621400
77652889	750100
69887600	876980
62122311	1012570
54357022	1133860
46591733	1263210
38826445	1387490
31061156	1520610
23295867	1648140
15530578	1775930
0	1902680

In the preceding example, note that the NPT value in the first entry is the same as the NPT value in the last entries of the two forward-direction files, above. Also, the file offset value in the last entry of the twenty-one times/reverse is the same as the corresponding entry of the twenty-one times/forward file.

With the “Bambi” content prepared as shown above, use a command such as the following to store the “Bambi” content package in the MFS:

```
% smc_tar xf /dev/rmt/0
```

See Chapter 5 for a complete description of `smc_tar`.

---

## B.8 References

ISO 8824 - Information Processing Systems - Open Systems Interconnection - Specifications for Abstract Syntax Notation One (ASN.1), 1987.

ISO 8825 Information Processing Systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1) 1987.

CCITT Recommendation X.209 (1988): Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1). Technically aligned with ISO 8825 and ISO 8825/AD 1.



# Glossary

---

This glossary contains general, multimedia-related terms, as well as terms specific to the Sun MediaCenter product. Words and phrases in the latter category are indicated as such.

<b>access control list</b>	A list of users and the associated operations they are allowed to perform. An access control list is a feature of the Media Stream Manager (MSM).
<b>analog video</b>	Video based on a continuous electrical signal. Distinguished from digital video, which is based on the periodic sampling of discrete components of a signal.
<b>B-frame</b>	A highly compressed bi-directional frame used in <b>MPEG</b> encoding. B-frames allow MPEG files to be played in reverse and play a role in MPEG's higher compression than other video compression schemes.
<b>chrominance</b>	A matrix, block, or single sample representing one of the two color difference signals related to the primary colors in the manner defined in the bit stream. The symbols for chrominance are Cr and Cb. Chrominance is distinguished from <b>luminance</b> .
<b>CIF</b>	Standard for resolution based on the throughput speed of compact disc drives. At CD-drive throughput (1.15 MB/sec.), a true-color (24 bits/pixel) video clip playing at 30 frames/sec. has a resolution size of 352 by 240 pixels. Resolution effects image size, as the image appears on a video display.
<b>composite</b>	An analog video signal composed of both luminance and chrominance.
<b>component</b>	A matrix, block, or single sample from one of three matrices ( <b>luminance</b> and two <b>chrominance</b> ) that make up a picture.
<b>content package</b>	Required input format for the Sun MediaCenter server. A content package contains one <b>Table of Contents (TOC) file</b> , plus index and data files that make up a movie. The <code>smc_tar</code> utility installs a content package so that it is accessible to MSM clients, through the TOC file. See Chapter 2 "Preparing and Loading Content," for details on content packages.

<b>luminance</b>	A matrix, block, or single sample representing a monochrome representation of the signal. Luminance is related to the primary colors in the manner defined in the bit stream. The symbol for luminance is Y. Luminance is distinguished from <b>chrominance</b> .
<b>Media Stream Manager (MSM)</b>	An RPC-based API that provides users with a means of interacting with a media server without coming close to the details of media file storage and the scheduling and delivery of media streams. The MSM supports VCR-like operations such as play-a-title and fast-forward, and video-editing functions such as insert-a-title-in-a-playlist. The MSM is described in detail in the <i>Sun MediaCenter Server Programmer's Guide</i> .
<b>MPEG</b>	Moving Picture Experts Group. MPEG-1 and MPEG-2 are international standards for the compression used in the conversion of analog to digital video, including both video and audio data. With MPEG-1, the ratio of the storage size of compressed to uncompressed digital video is 1:100. MPEG-2 extends MPEG-1 standards; it is intended to meet the needs of television broadcast studios. MPEG-2 approximately doubles the compression available in MPEG-1.
<b>NTSC</b>	National Technical Standards Committee. An analog video-format standard used in the United States, Canada, Japan, and Mexico. The NTSC format specifies 525 scan lines and a frame rate of 30 frames/sec. NTSC, <b>PAL</b> , and <b>SECAM</b> are the three video formats used throughout the world.
<b>PAL</b>	Phase Alternation Line. An analog video-format standard used mostly in Western Europe. The PAL format specifies 625 scan lines and a frame rate of 25 frames/sec. PAL, NTSC, and <b>SECAM</b> are the three video formats used throughout the world.
<b>pixel</b>	picture element
<b>playlist</b>	In the <b>Media Stream Manager (MSM)</b> , a list of titles to be played, annotated with a relative time at which the title should be played.
<b>play position</b>	In a <b>Media Stream Manager (MSM)</b> environment, the current position within a playlist from which isochronous data is being delivered.
<b>SECAM</b>	Sequential Couleur A Memoire. An analog video-format standard used in France, Russia, and Eastern Europe. As with the PAL format, SECAM specifies 625 scan lines and a frame rate of 25 frames/sec. However, SECAM encodes video information differently from PAL encoding. SECAM, NTSC, and <b>PAL</b> are the three video formats used throughout the world.
<b>session</b>	A period of interaction with a server delimited by the <b>Media Stream Manager (MSM)</b> functions <code>msmServerOpen()</code> and <code>msmServerClose()</code> .
<b>stream</b>	The representation of the delivery of an isochronous bit stream to a destination.



## **Table of Contents**

<b>(TOC) file</b>	A fundamental component of a <b>content package</b> . A TOC file lists the set of index and data files that contain the bit streams that make up the movie contained by that content package. <b>MSM</b> clients play movies by specifying a TOC file for that movie. See Section B.2 “The TOC File” on page B-3 for a complete description.
<b>trick play</b>	The capability for “VCR-like” functions of fast forward, fast reverse, and pause.
<b>wall-clock time</b>	Time of day. In a <b>Media Stream Manager (MSM)</b> environment, represented as seconds after 00:00:00 UTC Jan 1, 1970.



# Index

---

## A

- access control list, definition of, Glossary-1
- access control list, title, 5-1
- active streams, detecting number of, 3-21, A-2
- admin permission, requirement for one user in title ACL, 5-3
- Admission\_Failed trap, A-9
- alternatives for video file transfer, 2-5
- analog video, definition of, Glossary-1
- ASN.1, references to standards, B-18
- asterisk wildcard
  - limitations in use, 6-6
- asterisk wildcard, requirement for escaping, 6-6
- ATM addressing, 7-4
- ATM addressing over Classical IP, 7-4
- ATM and Fast Ethernet links, differences
  - between, 7-9
- ATM encapsulation of output data, 7-2
- ATM interfaces
  - addressing individually, 10-3
- ATM output environment, 1-9
- ATM virtual circuit, specifying, 7-4
- attributes, required for content loading, 6-14
- auto-detection of stream format, 5-5

## B

- backing up content, with smc\_copy, 5-4
- backing up files on system disk, 5-11
- backing up titles with smc\_tar, 5-11
- backup and restore, using ftp commands, 6-12
- backup utilities, 8-12
- B-frame, definition of, Glossary-1

- bit pump, 1-7

- bit pump characteristics, 3-23, A-3
- bit rate requirements for encoding, 2-2
- bit rate, encoding/play mismatch, 10-3
- bit rate, supported range for encoding, 2-2
- BitStreamObjects, portion of ContentObjects, B-6
- block number value, for smc\_tar, 5-8
- block size, recommended for smc\_tar, B-14

## C

- checking LEDs for troubleshooting, 8-10
- chrominance, definition of, Glossary-1
- CIF, definition of, Glossary-1
- closed\_gop flag, setting for trick play, 2-4
- component, definition of, Glossary-1
- composite, definition of, Glossary-1
- constant bit rate requirement, 2-2
- content encoding requirements, 2-1
- content files, removing, 5-21
- content input specification, 2-1, B-1
- content loading, 2-5
- content loading with ftp, 6-14
- content loading with smc\_tar, 5-9
- content loading, preferred method, 2-5
- Content Manager, introduction to, 1-8
- Content Manager, utilities that use .., 1-10
- content package, copying with smc\_copy, 5-4
- content package, definition of, Glossary-1
- content package, example of, B-14
- content package, introduction to, B-2
- content package, relationship to title, 5-1
- content package, when .. is required, B-1

- content requirements, 2-1
- content requirements for trick play, 2-2
- content, utility for listing, 5-20
- ContentObjects, ASN.1 description of content package, B-5
- copying content with smc\_tar, 5-10
- corrupt disk label, 10-3

## D

- data disk failure, 8-10
- DataFileObjects, portion of content object, B-7
- delete command, in ftp, 6-14
- design benefits, 1-12
- dir command, in ftp, 6-7
- Discover, using to find Sun MediaCenter server, A-8
- disk errors, MIB attribute for, 3-22, A-3
- disk failure, 8-10
- disk failure, following reboot, 8-7, 8-12
- disk failure, forcing .. with mfs repair, 8-8
- disk failure, parity disk, 8-12
- disk failure, recovering from, 8-5
- disk failure, smc\_tar response to, 5-10
- Disk I/O, MIB attributes for, A-6
- disk replacement, 1-6, 8-7
- disk space
  - checking available .. in MFS, 8-2
- disk striping used in Media File System, 1-6
- Disk\_Failure SNMP trap, 8-2
- Disk\_Failure trap, A-10

## E

- encapsulation, specifying for output, 7-8
- encoding requirements, 2-2
- enterprise-specific traps, A-9
- error messages, 10-4
- Ethernet address, requirement for destination clients, 10-2

## F

- failed disk replacement, 1-6
- failure, single-disk, 8-10
- Fast Ethernet addressing, 7-7

- Fast Ethernet and ATM links, differences
  - between, 7-9
- Fast Ethernet data packet format, 7-5
- Fast Ethernet output data encapsulation, 7-5
- FastEthernet interface
  - treatment of multiple, 10-3
- file backup, system-disk files, 5-11
- file locking, as implemented by the MFS, 1-6
- file offset, field in index file, B-12
- file size limit, MFS, 1-6
- file size, checking on, 8-5
- filesystem size limit, MFS, 1-6
- filesystem, remaking, 8-12
- format attribute, for ftp, 6-4
- format detection, relying upon .. in smc\_copy command, 5-5
- ftp client commands, 6-5
- FTP daemon, for the Sun MediaCenter server, 6-1

## G

- get command, in ftp, 6-11

## H

- hot swappability, 1-6

## I

- index file, automatic generation for MPEGTS, 6-15
- index file, automatic generation for MPEGTS streams, 2-6
- index file, description of and requirements for, B-10
- index file, stream switching in the absence of, 2-3

## J

- jitter and drift, defined limits to, 1-12

## L

- LAN environment, 1-9
- large MPEG files, building TOC file for, B-9
- listing MFS files, utility for, 5-17
- loading content with ftp, 6-14

loading content with smc\_tar, 5-9  
loading content, with smc\_copy, 5-4  
ls command, in ftp, 6-8  
luminance, definition of, Glossary-2

## **M**

management database, saving in Solstice Domain Manager, A-8  
Media File System  
    creation of, 8-8  
Media File System (MFS)  
    features of, 1-5  
Media File System bit pump, 1-7  
Media File System output driver, 1-7  
Media File System subsystems, 1-6  
Media Stream Manager (MSM), definition of, Glossary-2  
Media Stream Manager (MSM), use of, 5-1  
Media Stream Manager, introduction to, 1-7  
Media Transmission Unit for ATM, 7-3  
MFS  
    creation of, 8-8  
    remaking filesystem, 8-12  
mfs df utility, 8-2  
mfs diskusg utility, 8-4  
MFS errors, displaying, 10-5  
MFS file size limit, 1-6  
MFS meta-file, location of, 8-10  
mfs perror utility, 10-5  
mfs repair utility, 8-5  
MFS size limit, 1-6  
MFS space requirements in /var, 2-6  
MFS-to-UFS file copy, 5-4  
mget and mput, limitations in use, 6-12  
mget command, in ftp, 6-12  
MIB for SMC server, A-2  
Missed\_Deadline trap, A-10  
mkmfms utility, 8-8  
mls command, in ftp, 6-8  
monitoring MFS disks, 8-2  
MPEG encapsulation over Classical IP over AAL5, 7-3  
MPEG encoding requirements, 2-2  
MPEG encoding requirements for content, 2-1  
MPEG encoding requirements for trick play, 2-2  
MPEG, definition of, Glossary-2  
MPEG-1 System Stream encapsulation for ATM, 7-3

MPEG-1 System Stream encapsulation for Fast Ethernet, 7-7  
MPEG-2 Program Stream encapsulation for Fast Ethernet, 7-7  
MPEG-2 Program Stream packet encapsulation for ATM, 7-3  
MPEG-2 Transport Stream encapsulation for Fast Ethernet, 7-6  
MPEG-2 Transport Stream packet encapsulation for ATM, 7-2  
mput command, in ftp, 6-13  
MsmConnect structure, encapsulation field in, 7-8  
MsmServer process, 10-2  
MsmSetConnect() call, to specify encapsulation, 7-8  
multiple disk failures, 8-12  
multiple Fast Ethernet links, treatment of, 7-9  
multiple titles, moving using ftp, 6-12

## **N**

name attribute, for ftp, 6-4  
naming convention for titles, B-5  
NFS file, loading content from, 5-7  
Normal Play Time, field in index file, B-11  
normal speed, value used for, 5-20  
NTSC, definition of, Glossary-2

## **O**

online loading, 1-5  
online loading, MIB attribute for, 3-21, A-4  
output data format, 7-1  
output driver, 1-7

## **P**

PAL, definition of, Glossary-2  
parity disk, restoring, 8-12  
parity mechanism, protection against single-disk failure, 8-10  
path, to MFS utilities, 8-2  
perror utility, 10-5  
pixel, definition of, Glossary-2  
play position, definition of, Glossary-2  
playlist, definition of, Glossary-2  
playthrough feature, 2-6

playthrough, support in the MFS, 1-5  
Program Association Table, encoding  
    requirement, 2-2  
Program Clock Reference, encoding  
    requirement, 2-2  
push model, description of, 1-11  
put command, in ftp, 6-11

## R

RAID level supported by MFS, 1-2  
random\_access\_indicator, setting for trick play, 2-4  
rate attribute, for ftp, 6-4  
recovery from single-disk failures, 1-5  
references, used for content storage  
    requirements, 2-7  
remote copying and extraction of titles, 5-12  
remove utility, 5-21  
rename command, in ftp, 6-13  
restoring content, 8-10, 8-12  
RPC server process, 10-2

## S

SCSI target numbers, as used in mkmfs, 8-10  
SECAM, definition of, Glossary-2  
server output data format, 7-1  
server processes, 10-2  
session (MSM), definition of, Glossary-2  
smc  
    keyword for ftp, 6-3  
smc\_copy utility, 5-4  
smc\_gettacl utility, 5-3  
smc\_ls utility, 5-17  
smc\_rm utility, 5-21  
smc\_settacl utility, 5-2  
smc\_tar command, in pipeline with tar  
    command, 5-13  
smc\_tar command, to access file on CD-ROM, 5-14  
smc\_tar utility, description of, 5-8  
smc\_tar, c (create) option, 5-8  
smc\_tar, x (extract) option, 5-8  
SNMP agent for the video server, 8-2  
SNMP agent, instructions for configuration, A-1  
SNMP agent, limitation to video server MIB, A-2  
SNMP MIB for server, A-2  
SNMP traps, enabling handling of, A-9

snmp.traps file, A-9  
snmpd daemon, A-2  
snoop, used for troubleshooting, 10-1  
Solstice Domain Manager, using with SMC  
    MIB, A-6  
space consumption in /var, 2-6  
SPARCstorage MultiPack enclosure, monitoring  
    LEDs, 8-10  
SPARCstorage RSM trays, 8-10  
speed attribute, for ftp, 6-4  
splice point, description of and requirements  
    for, 2-3  
splice points, in an index file, B-11  
stream admission, MIB attributes for, A-6  
stream delivery by Media File System, 1-5  
stream format, specifying in smc\_copy  
    command, 5-5  
stream, definition of, Glossary-2  
Sun Media Center hardware, 1-2  
Sun Media Center software  
    overview of, 1-5  
Sun MediaCenter FTP daemon, 6-1  
Sun MediaCenter server  
    design goals, 1-2  
Sun MediaCenter Server MIB, A-2  
Sun MediaCenter server MIB, A-8  
SunNet Manager, using with SNMP agent, A-6  
supported formats, for content loading with  
    FTP, 6-14

## T

Table of Contents (TOC) file, definition  
    of, Glossary-3  
Table of Contents (TOC) file, description of, B-3  
Table of Contents (TOC) file, order of index and data  
    files within, B-13  
tape sizes supported, B-14  
tape storage format, required for server, B-12  
tar file, copying to/from a Sun MediaCenter  
    server, 5-4  
tar file, creation with smc\_copy, 5-7  
tar format, recommended block size, B-14  
third-party copy operation, 5-6, 5-7  
Thomson Consumer Electronics packet  
    encapsulation for ATM, 7-3  
Thomson Consumer Electronics packet  
    encapsulation for Fast Ethernet, 7-7

- title access control list, 5-1
- title ACL permissions, 5-2
- title naming convention, B-5
- title, setting access to, 5-2
- titles, obtaining list of on server, 5-17
- titles, removing, 5-21
- TOC file for large MPEG files, B-9
- TOC file, example for simplest case, B-8
- TOC file, parsing, 5-9
- trick play streams, loading, 6-15
- trick play, definition of, Glossary-3
- troubleshooting, 10-1
- type attribute, for ftp, 6-4

## **U**

- underperforming stream output, 10-2
- Unix filesystem, copying to/from using
  - smc\_copy, 5-5

## **V**

- video file attributes, 6-4
- video file transfer, alternatives for, 2-5
- video file usage, detecting, 5-20
- video file, copy operations, 5-6

## **W**

- wall-clock time, definition of, Glossary-3
- Web-based tools, introduction to, 1-8
- wildcard
  - limitations for use with Sun MediaCenter FTP daemon, 6-6

